ANSI X9.17-1985
(Reaffirmed 1991)

AMERICAN NATIONAL STANDARD

Financial Institution Key Management (Wholesale)

DEVELOPED BY
ACCREDITED STANDARDS COMMITTEE
X9-FINANCIAL SERVICES
PUBLISHED BY
AMERICAN BANKERS ASSOCIATION
X9-SECRETARIAT

Financial Institution Key Management (Wholesale)

X9.17

Approved April 4, 1985

Developed by the Accredited Standards Committee on Financial Services, X9, operating under the procedures of the American National Standards Institute.

Published by the X9 Secretariat, American Bankers Association, 1120 Connecticut Avenue, N.W., Washington, D.C. 20036.

AMERICAN BANKERS ASSOCIATION
1120 Connecticut Avenue, N.W.
Washington, D.C. 20036

ABSTRACT

Today, financial institutions transfer trillions of dollars in funds and securities by electronic means. Sophisticated data security is required. This key management standard, utilized in conjunction with the American National Standards Institute Data Encryption Algorithm (DEA), can be used to protect financial messages and other sensitive information. Since the Data Encryption Algorithm is in the public domain, the security and reliability of any process based on the DEA is directly dependent on the protection afforded to secret numbers called keys. This standard provides a uniform process for the protection and exchange of these cryptographic keys for authentication and encryption. To provide this security, this standard establishes methods (including the protocol) for the generation, exchange, use, storage and destruction of these secret keys. This standard not only permits interoperability among financial institutions, but also permits interoperability between financial institutions and their wholesale customers. The logistics of traditional, manual key distribution methods are time consuming, expensive and insecure. By automating the process, the complexity and associated costs are reduced and security is dramatically improved.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

Foreword

Financial institutions are making increased use of the American National Standard Data Encryption Algorithm, DEA[1], to protect financial messages and other sensitive information. Specific examples of this include message encryption and funds transfer message authentication.

The DEA algorithm is in the public domain. The security and reliability of any process based on the DEA is directly dependent on the protection afforded to a secret number, called the key. To provide this security, this standard establishes methods for the generation, distribution, storage and destruction of the secret key and other related information.

A familiar analogy may be found in the combination lock of a vault. The lock design is public knowledge. Security is provided by keeping a number, the combination, secret. Secure operation also depends on protective procedures and features which prevent surreptitious viewing or determination of the combination by listening to its operation. Procedures are also required to ensure that the combination is random and cannot be modified by an unauthorized individual without detection.

To support the varying needs of financial institutions and to support interoperability, this standard defines both manual and automated methods for exchanging keying material.

The key(s) must be:

- randomly generated and distributed, stored, and destroyed or archived in a secure, controlled, and auditable manner.

- tested during system operation to detect system failures or unauthorized changes.

Implementation of these requirements will also require certain system tests, generation of appropriate alarms and inhibition of operations that could result in compromising the key, the data or both.

While the techniques specified in this standard are designed to maintain the security and integrity of keys, the standard in no way guarantees that a particular implementation of the techniques is secure.

---

[1] ANSI X3.92-1981, Data Encryption Algorithm.

It is the responsibility of the financial institution to put overall key management in place with the necessary controls to assure that the process is implemented under secure procedures. Further, the process should be audited to ensure compliance with the procedures.

Suggestions for the improvement or revision of this standard will be welcome. They should be sent to the X9 Committee Secretariat, American Bankers Association, 1120 Connecticut Avenue, N.W., Washington, D.C. 20036.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Financial Services-X9. Committee approval of the standard does not necessarily imply that all the committee members voted for its approval. At the time it approved this standard, the X9 Committee had the following members:

Robert Kaminski, Chairman
Donald Monks, Vice Chairman
Cynthia Fuller, Secretariat

Organization Represented	Representative
American Express Company	Bonnie Howard
Bank of America	Russel Fenwick
Burroughs Corporation	Stanley Fenner
Chase Manhattan Bank, NA	James Zegion
Chemical Bank	W. Robert Moore
Citibank	Seymour Rosen
Continental Illinois National Bank & Trust	Joseph Coriaci
Cuna Services Group	Martha Reed
Depository Trust Company	Janet Rader
Digital Equipment Corporation	Donald B. Holden
Dollar Dry Dock Savings Bank	John Petrusky
Eastman Kodak Company	Eva Martin
Federal Reserve of Dallas	Johnny Johnson
First Interstate Services	Charles Pecharka
Honeywell Information Systems, Inc.	Charles McGrory
IBM Corporation	Leighton Carmichael
Irving Trust Company	Donald Monks
MasterCard International	Neil Thompson
Mellon Bank	Eugene Cooney
Moore Business Forms, Inc.	Delmer Oddy
NCR Corporation	A.R. Daniels
Recognition Equipment, Inc	Travis Hammer
Security Pacific National Bank	B. Jean Christy
UARCO, Inc.	Lois Richards

U.S. League of Savings Institutions	O. Tom Thomas
Valley National Bank	Gene Saunders
VISA, USA	Jean McKenna

The X9E9 Working Group had the following members:

Organization Represented	Representative
Analytics Communications Systems	Donald A. Cole
	Thomas Mitchell
Atalla Technovations	Dale Hopkins
	Frank Piedad
Bank Administration Institute	Thomas Tucker
Bank of America	Glenda Barnes
	Suzanne Gray
BANKWIRE	Barbara Davis
Chase Manhattan Bank	James Siler
	Richard Yen
Citibank	M. Blake Greenlee
Citicorp, TTI	Grant Laney
COMPLAN	Herbert Bright
	Richard Enison
Consultant	Howard Zeidler
Department of the Treasury	Richard Bauder
	Martin Ferris
Federal Reserve Bank of New York	Howard Crumb
Federal Reserve Bank of Richmond	William Glover
IBM Corporation	Robert Elander
	Richard Lennon
Jones Futurex	Edward Grundler
Manufacturers Hanover Trust	Howard Peace
Mellon Bank, N.A.	Martin Matthews
	Robert Pelto
	David Taddeo
National Bureau of Standards	Elaine Barker
	Miles Smid
National Security Agency	Joseph Mettle
	Gerard Rainville
Paradyne	Steven Maxwell
	R.K. Smith
PE Systems	William Barker
	John Romine, Jr.
Profile Analysis Corporation	Peter S. Browne
Racal-Milgo, Inc.	Richard Abruscato
	Laura Schlafly

Security Pacific Bank	Barbara Leider
	Arthur Nelson
	Edward Zeitler
Total Assets Protection, Inc.	Ralph Spencer Poore

1. Scope

This standard covers both the manual and automated management of keying material for the wholesale financial services industry. This standard specifies the minimum requirements for the management of keying material, including:

- Control during the life of the keying material to prevent unauthorized disclosure, modification or substitution.
- Distribution of keying material to permit interoperability between cryptographic equipment or facilities.
- Ensuring the integrity of keying material during all phases of its life, including its generation, distribution, storage, entry, use, and destruction.
- Recovery in the event of failure of the key management process or when the integrity of the keying material is questioned.

2. Definitions and Common Abbreviations

2.1 Definitions

Authentication	The act of determining that a message has not been changed since leaving its point of origin. The identity of the originator is implicitly verified.
Biased	With respect to generation of random or pseudo-random numbers, a process is biased if the occurrence of some numbers is more likely than others.
Ciphertext	Encrypted (enciphered) data.
Communicating Pair	Two logical parties who have previously agreed to exchange data. A party and a center exchanging cryptographic service messages do not constitute a communicating pair.
Cryptoperiod	The time span during which a specific key is authorized for use or in which the keys for a given system may remain in effect.
Cryptographic Equipment	A device wherein cryptographic functions (e.g., encryption, authentication, key generation) are performed.
Cryptographic Key (Key)	A parameter that determines the transformation from plaintext to ciphertext and vice versa. (A DEA key is a 64-bit parameter consisting of 56 independent bits and eight bits which may be used as odd parity bits.)

Cryptographic Keying Material	See Keying Material.
Cryptographic Service Message	A message for transporting keys or related information used to control a keying relationship.
Data Encryption Algorithm (DEA)	The encryption algorithm specified by ANSI X3.92-1981, Data Encryption Algorithm.
Data Key	A key used to encrypt and decrypt, or to authenticate data.
DEA Device	The electronic hardware part or subassembly which implements only the DEA as specified in ANSI X3.92-1981, and which is validated by the National Bureau of Standards.
Decryption	A process of transforming ciphertext into plaintext.
Degauss	To remove, erase or clear information from magnetic media.
Dual Control	A process of utilizing two or more separate entities (usually persons), operating in concert, to protect sensitive functions or information. Both entities are equally responsible for the physical protection of materials involved in vulnerable transactions. No single person shall be able to access or to utilize the materials (e.g., cryptographic key). For manual key generation, conveyance, loading, storage and retrieval, dual control requires split knowledge of key among the entities. (Also see Split Knowledge)
Encryption	A process of transforming plaintext into ciphertext for the purpose of security or privacy.
Exclusive-or	A mathematical operation, symbol +, defined as: 0 + 0 = 0 0 + 1 = 1 1 + 0 = 1 and 1 + 1 = 0 Equivalent to binary addition, without carry.
Field Tag	A unique string of characters which identifies the meaning and location of the associated data field.
Financial Message	A communication containing information which has financial implications.

Initialization Vector (IV)	A number used as a starting point for encryption of a data sequence to increase security by introducing additional cryptographic variance and to synchronize cryptographic equipment.
Interoperability	The ability to exchange keys, both manually and in an automated environment, with any other party implementing this standard, providing that both implementations use compatible options of this standard and compatible communications facilities.
Key	See Cryptographic Key.
Key Component	One of at least two parameters having the format of a cryptographic key that is exclusive-or'ed with one or more like parameters to form a cryptographic key.
Key Encrypting Key	A key used exclusively to encrypt and decrypt keys.
Key Generator	A device, including associated alarms and self tests, for generating cryptographic keys (and where needed, IVs).
Key Loader	An electronic, self-contained unit which is capable of storing at least one key and transferring that key, upon request, into cryptographic equipment.
Key Management Facility	The physically protected enclosure (e.g., room or device) and its contents where cryptographic elements (i.e., cryptographic hardware, software, firmware, keys, or IVs) reside.
Keying Material	The data (e.g., keys and IVs) necessary to establish and maintain cryptographic keying relationships.
Key Offset (Offset)	The process of exclusive-or'ing a counter to a key.
Keying Relationship	The state existing between a communicating pair during which time they share at least one data key or key encrypting key.
Logical Party	One or more physical parties that form one member of a communicating pair.
Message	A communication containing one or more transactions or related information.
Message Authentication Code (MAC)	A number which is the result of passing a message through the authentication algorithm using a specific key. Reference ANSI X9.9-1982, Financial Institution Message Authentication (Wholesale).

Notarization	A method of applying additional security to a key utilizing the identities of the originator and the ultimate recipient.
Offset	See Key Offset.
Optional	Not required by this standard or not required to meet an optional provision of this standard.
Originator	The person, institution or other entity that is responsible for and authorized to originate a message.
Plaintext	Unencrypted (unenciphered) data.
Recipient	The person, institution or other entity that is responsible for and authorized to receive a message.
Security Life	The time span over which cryptographically protected data have value.
Split Knowledge	A condition under which two or more parties separately have key components which, individually, convey no knowledge of the resultant cryptographic key. The resultant key exists only within secure equipment as the automatically generated, exclusive-or'ed result of the full length key components which each individual entered separately and confidentially.
Zeroization	A method of degaussing, erasing or overwriting electronically stored data (see Appendix E).

2.2 Common Abbreviations

This section contains abbreviations commonly used in this standard.

ABBREVIATION	MEANING	REMARKS
CKD	Key Distribution Center	A facility which generates and returns keys for distribution.
CKT	Key Translation Center	A facility which transforms and returns keys for distribution.
CSM	Cryptographic Service Message	Tag for cryptographic service messages.

ABBREVIATION	MEANING	REMARKS
CTA	Count, "A"	An incrementing binary counter used to control successive key distributions under a particular key encrypting key. Used between a Key Distribution Center or a Key Translation Center and a party designated as "A".
CTB	Count, "B"	An incrementing binary counter used to control successive key distributions under a particular key encrypting key. Used between a Key Distribution Center or a Key Translation Center and a party designated as "B".
CTP	Count, "P"	An incrementing binary counter used to control successive key distributions under a particular key encrypting key. Used in a point-to-point relationship.
CTR	Count, "R"	The count field of an error message which is equal to the received count and is sent only when a count error occurs.
DSM	Disconnect Service Message	Optional message class used to discontinue one or more keys or to terminate a keying relationship.
EDC	Error Detection Code	An error detection code generated using the authentication algorithm and the fixed hexadecimal key, 0123456789ABCDEF.
EDK	Effective Date of Key	Date and Coordinated Universal Time when the data key shall be activated.
ERF	Error Field	The field which identifies error conditions detected in a prior Cryptographic Service Message.
ERS	Error Recovery Service Message	Used to recover from count or other errors in a Key Distribution Center or Key Translation Center environment.
ESM	Error Service Message	Used to give a negative acknowledgment on receipt of any Cryptographic Service Message other than an ESM and to give the recipient data with which to recover.

ABBREVIATION	MEANING	REMARKS
IDA	Identity of Key for Authentication	Identifies the key to be used to authenticate a Disconnect Service Message. This key shall be discontinued.
IDC	Identity of Key Distribution Center or Key Translation Center	
IDD	Identity of key to be discontinued	
IDK1	Key Identifier (subfield)	Identifies (names) the key being sent in a Cryptographic Service Message key field.
IDK2	Key Encrypting Key Identifier (subfield)	Identifies (names) the key encrypting key or key pair used to encrypt the key being sent in a Cryptographic Service Message key field.
IDU	Identity of Ultimate Recipient	This field is only used with a Key Distribution Center or a Key Translation Center.
IV	Initialization Vector	Starting point for a DEA encryption/decryption process.
KD	Data Key	
KDU	Data Key, Notarized	A data key for the ultimate recipient encrypted under a notarizing key.
KK	Key Encrypting Key	
*KK	Key Encrypting Key Pair	Consists of two key encrypting keys used together to encrypt other keys.
KKU	Key Encrypting Key, Notarized	A key encrypting key for the ultimate recipient, encrypted under a notarizing key.

ABBREVIATION	MEANING	REMARKS
*KKU	Key Encrypting Key Pair, Notarized	A key encrypting key pair for the ultimate recipient, encrypted under a notarizing key.
KSM	Key Service Message	Used to transfer keys between communicating pairs.
MAC	Message Authentication Code	
MCL	Message Class	The tag for the field that defines the type of Cryptographic Service Message.
NOS	Notarization Indicator	A tag that, when present, indicates that notarization was used.
ORG	Originator	Identity of the Cryptographic Service Message originator.
P	Key Parity (subfield)	Indicates that the plaintext key conforms to the specification for odd parity.
RCV	Recipient	Identity of the Cryptographic Service Message recipient.
RFS	Request For Service Message	Used to request translation of keys by a Key Translation Center for retransmission to another party.
RSI	Request Service Initiation Message	Optionally used to request keys from another party.
RSM	Response Service Message	Used to provide an authenticated acknowledgment.
RTR	Response To Request Message	Used to send keys from a Key Distribution Center or from a Key Translation Center.
SVR	Service Request	Specifies type of service requested.

3. Key Management Facility Protection

3.1 General

This section prescribes the standard level of protection which shall be present to assure the security of the keying material and the integrity of the key management facility.

3.2 Facility Requirements

The key management facility shall be designed so as to protect its contents from unauthorized disclosure, modification, substitution, insertion, and deletion.

The level of protection shall be such that unauthorized attempts to access the contents of the key management facility will either be unsuccessful or have a high probability of being detected and reported.

All cryptographic equipment shall reside within a key management facility. Cryptographic equipment may itself be a key management facility, in which case it would provide all the access controls required of a key management facility.

3.3 Key Entry

The key management facility shall permit, at either the system level or the device level, the capability of entering keys in a format conforming to this standard.

3.3.1 Manual Entry

A means shall be provided for the manual entry of human-readable keys. Manual entry of plaintext keys shall be accomplished under dual control, which requires split knowledge (refer to Section 2.1). If any plaintext key component is displayed, it shall be visible only to authorized personnel and shall be cleared immediately after the key entry process is completed. A means of correcting individual component errors or of entering the entire key shall be provided.

3.3.2 Automated Entry

If automatic entry is provided, electronically entered keys shall not be displayed during the entry process.

3.3.3 Protection

Access to key entry controls or systems shall be limited by logical or physical means, or both. Access shall be available only to authorized personnel.

3.3.4 Parity Checking

The odd parity of keys or key components generated in plaintext form shall be verified during key entry to preclude accidental single-bit key modifications.

3.4 Transportation and Storage

Keying material shall be transported and stored in such a manner as to preclude modification or substitution and to prevent disclosure of plaintext keys before, during or after the period in which the keys are active.

Storage shall be tamper resistant. If unauthorized access is attempted, the attempt shall have a high probability of being detected.

Access to physical storage shall be under dual control with full accountability. When keying material is entered or removed, the physical access shall be specifically authorized, physically or logically constrained, and fully documented.

3.4.1 Manual Transmittal to and from Storage

The movement of any keying material to or from storage shall be made under dual control with full accountability.

3.4.2 Storage of Active Keying Material

After entry into cryptographic equipment, any intermediate storage of plaintext keys shall be temporary and the keys shall be zeroized upon transfer of the keys to another location.

Except when being output from a key generation facility, keys shall never be available in plaintext form from any cryptographic equipment, even upon failure.

A means shall be provided for manually initiated zeroization of plaintext keys.

3.5 Cryptographic Equipment Operational Integrity

Cryptographic equipment shall be designed so as to minimize errors, to detect failure, and to minimize the possibility of key compromise.

3.5.1 Tests

At the time of manual key entry and system initialization, a functional test shall be performed to assure correct operation. A recommended approach is in Appendix D, Sections D.3 and D.4. Cryptographic equipment shall detect and report the erasure, loss or lowering of a counter, when used (see Section 7.3).

Key generator output shall be tested, when used, for generator failure (e.g., the repeated generation of the same key). Detection of such failure(s) shall cause inhibition of operation. Errors shall be documented and reported (manually or automatically).

3.5.2 Control Functions

Cryptographic equipment shall be designed with sufficient control functions to assure proper operation. See Appendix D, Section D.5, for a list of recommended control functions.

3.5.3 Indicators and Alarms

Cryptographic equipment shall be implemented such that there is an indication upon failure or error. For device implementations, sample indicators are described in Appendix D, Section D.5.

3.5.4 Electromagnetic Interference (EMI) Requirements

Good engineering design practice, to minimize the chances of compromise of key through radiation or conduction, shall be employed in the design of both cryptographic equipment and key loaders.

Appendix D, Section D.7 describes recommended requirements and construction practices for the reduction of EMI.

3.6 Destruction of Keys

Except for archival purposes, when keys have been compromised, suspected of having been compromised, or discontinued, they shall be physically or logically destroyed.

A key used by a communicating pair and discontinued by either of the pair shall not be intentionally re-used for any subsequent connections.

3.6.1 Method of Destruction

Paper-based keying materials shall be destroyed by crosscut shredding, burning or pulping. Keying material stored on other media shall be destroyed so that it is impossible to recover by physical or electronic means. Appendix E contains suggested methods for the destruction of keying materials.

3.6.2 Accountability

Destruction of keys shall be accomplished under conditions of full accountability, with appropriate records retained for audit trail purposes.

3.6.3 Archived Keys

If reconstruction of a given key and its related message is required at a later date, the keys shall be retained in such a form as to preclude their being intentionally used again as keying material.

Keys retained for message reconstruction purposes shall be identified (or converted into a form or format) so that there is no ambiguity regarding the fact that they are obsolete.

4. Key Management Requirements

4.1 General Requirements

The functions of key management are to provide keys and Initialization Vectors (IVs) where they are needed and to keep keys secret. The security of the encrypted or authenticated data is strictly dependent upon the prevention of unauthorized disclosure, modification, substitution, insertion, and deletion of keys or IVs. If these are compromised, the security of the data can no longer be assured. When the physical parties share a common key, the ability to distinguish cryptographically between the physical parties is not provided by this standard. In all cases, the manual distribution of keys shall be physically protected.

a. Disclosure

Secret cryptographic keying material (i.e., keys) shall be protected from unauthorized disclosure throughout the security life of the data which it protects. This protection shall be provided by either physical protection or encryption.

Note: When secret cryptographic keying material is sent over a communications channel which is not physically secured (e.g., a telephone line), then encryption shall be used to prevent disclosure.

b. Modification

All cryptographic keying material, secret and non-secret, shall be protected to prevent or detect any unauthorized modifications. This protection shall be provided by either physical security or cryptographic authentication.

Note: When cryptographic keying material is sent over a communications channel which is not physically secured, modification cannot be prevented. Modification shall be detected by means of the cryptographic authentication algorithm defined in ANSI X9.9-1982.

c. Substitution, Insertion, and Deletion

Where protection by physical security is not possible, all cryptographic keying material shall be protected from substitution, insertion, and deletion by the use of a counter (see Section 7.3) in conjunction with cryptographic authentication.

4.2 Requirements for the Automated Key Management Architecture

The following list of requirements assumes that:

- the data network shall be expandable.
- either a communicating pair has a key encrypting key in common or each has a key encrypting key pair in common with a center.

Figure I provides a pictorial representation of the architecture.

(1) The architecture shall support the ability to have at least one data key between communicating pairs.

(2) The architecture shall support the ability to change data keys automatically between communicating pairs.

(3) A data key can be used for either encryption or authentication but not both, except for a Cryptographic Service Message.

(4) A data key or key encrypting key shared between a communicating pair shall not be disclosed to a third party (except for a Key Translation Center (CKT) or a Key Distribution Center (CKD)).

(5) A data key shared between a communicating pair shall be secured from third party usage (except for a CKD or CKT).

(6) The compromise of any key shared between any communicating pair shall not compromise any third party.

(7) The architecture shall support communicating parties that do not have a key generation capability.

(8) Data keys may be sent for present or future use.

(9) Key security and integrity shall be ensured.

(10) The architecture shall support any party initiating a secure connection with any other party.

(11) A key used between any communicating pair shall not intentionally be used between any other communicating pair.

(12) A key that has been used between a communicating pair and discontinued by either of the pair shall not be intentionally re-used for any subsequent connection.

(13) A means shall be provided for either party to discontinue a key.

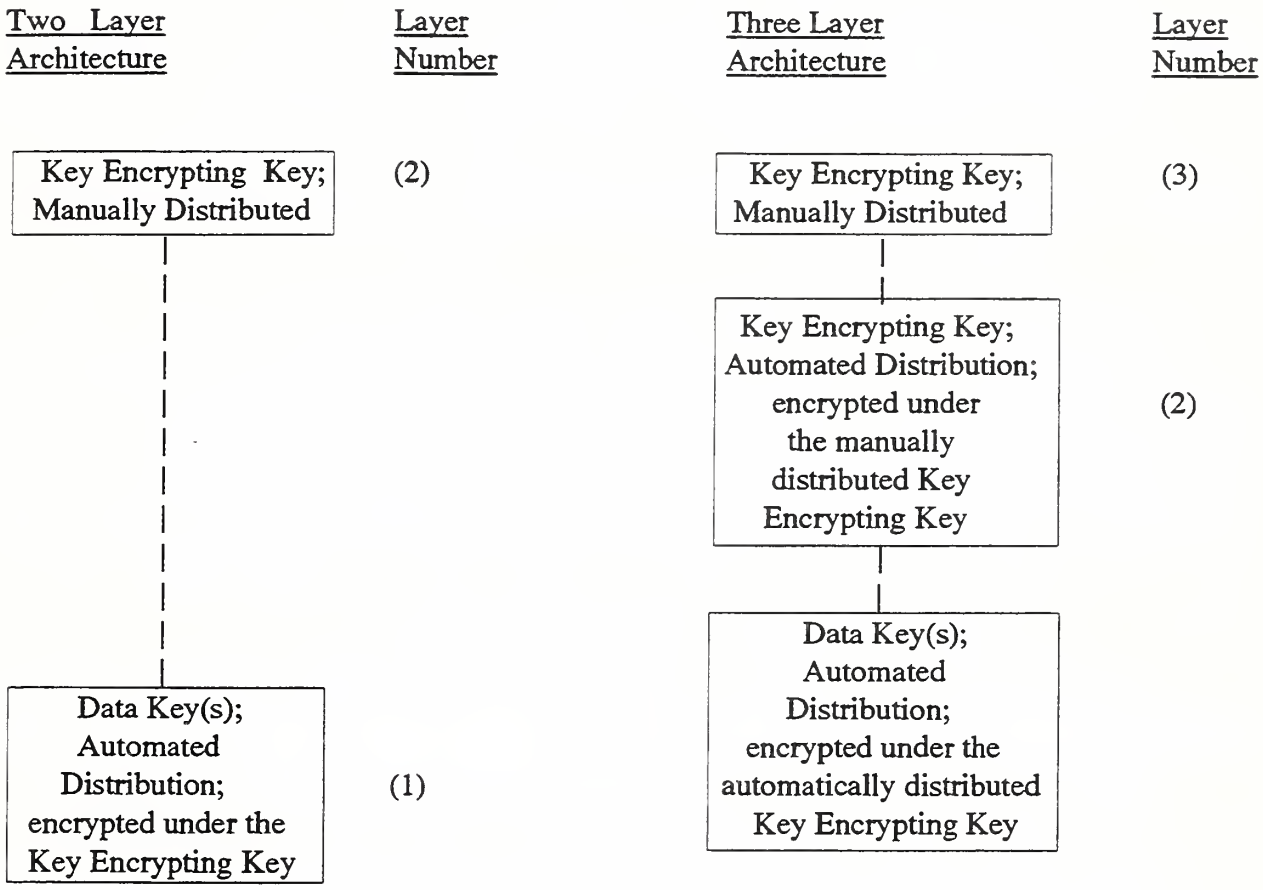
(14) The architecture shall permit 2 or 3 layers of keys (see section 4.3).

(15) Any communicating pair may share more than one key encrypting key.

(16) The ability to exchange key encrypting keys automatically between pairs in a 3 layer architecture shall be provided.

FIGURE I

Key Distribution Architecture



4.3 Automated Key Management Architecture

The architecture shall consist of either two or three layers of keys. All implementations shall have the capability of functioning in a two layer architecture.

In a two layer architecture, a manually distributed key encrypting key shall be used to encrypt data keys for distribution.

In a three layer architecture, automatically distributed key encrypting keys shall be encrypted using a manually distributed key encrypting key. One or two data keys shall be transmitted with the automatically distributed key encrypting key. These data keys shall be encrypted under the automatically distributed key encrypting key. Subsequent key distribution messages need not include a key encrypting key. When no key encrypting key is transmitted, one or two data keys shall be sent and shall be encrypted under an automatically distributed key encrypting key which has been previously exchanged between the communicating pair.

The highest layer key encrypting key(s) shall be manually distributed. Manually distributed keys shall never be electronically transmitted. Key encrypting keys shared with a Key Distribution Center (CKD) or a Key Translation Center (CKT) shall consist of key pairs. Other communicating pairs may share either single key encrypting keys or key encrypting key pairs. Manually distributed key encrypting keys shall not be superseded except by other manually distributed key encrypting keys. Parties which are not centers shall only accept key encrypting keys encrypted under manually distributed key encrypting keys.

The transmitted key encrypting keys and data keys may be used in addition to or may supersede key encrypting keys and data keys, respectively, previously exchanged electronically.

5. Key and Initialization Vector Generation

5.1 General Requirements

Keys and Initialization Vectors (IVs) shall be generated so that keys and IVs are random or pseudorandom. On each key or IV generation, every key or IV in the space of all possible keys or IVs shall have an equal chance of being selected and shall have no apparent relationship to its predecessor or successor. The strength of the key generation process shall be at least equal to the strength of the resulting keys to be used. The key generation process shall be designed so that no cryptographic advantage is gained by attacking the key generation process rather than the encryption algorithm. Appendix C describes a method for generating keys and IVs.

5.2 Generation of Keys and IVs for Manual Distribution

All key generation, distribution and storage resources (e.g., copies, ribbons, etc.) shall be protected from unauthorized use, alteration, replacement, destruction or exposure. Waste products shall be destroyed. The key generation process shall take place in a secure area to which access is controlled and where unauthorized viewing is prevented.

When keys or IVs are printed, provision shall be made to protect the keying material from unauthorized disclosure or replacement. Such protection could include uniquely identified key books and numbered pages protected by tamper resistant packaging.

Key and IV generation procedures shall be under dual control.

5.3 Generation of Keys and IVs for Automated Distribution

All automated resources which generate keys and IVs shall be physically protected to prevent the:

(1) disclosure, modification and replacement of the keys,

(2) modification or replacement of the IVs,

(3) modification or replacement of the generation algorithm, or device.

6. Key and IV Distribution

6.1 General

Keys, and where needed, IVs, shall be distributed to all key management facilities before secure transactions can take place and such distribution shall be done in a secure manner.

In no case shall keys be used prior to the receipt of valid acknowledgments or if compromise is suspected. Procedures to follow up and resolve distribution irregularities shall be in place.

6.2 Automated Key Distribution Environments

Three environments exist for key distribution: point-to-point, Key Distribution Center (CKD) and Key Translation Center (CKT).

(1) A Point-to-Point Environment

A point-to-point environment exists when two parties (a communicating pair) share a key encrypting key, either a single key or a key pair, so that further key encrypting keys and data keys may be exchanged. At least one of the parties shall have the capability to generate or otherwise acquire keys. The implementation of the point-to-point environment is the minimum requirement of this standard for automated key distribution.

(2) A Key Distribution Center Environment

A Key Distribution Center exists for the purpose of distributing generated or acquired data keys to two parties who:

(a) wish to communicate with each other but do not currently share keys,

(b) each share a key encrypting key pair with the Key Distribution Center,

(c) may not have the ability to generate keys.

One of the parties (the originator) requests data key(s) from the Key Distribution Center for later communication to the other party (the ultimate recipient).

The Key Distribution Center generates or acquires the data key(s) and sends two identical sets to the originator using a key encrypting key pair shared with the originator to key offset and encrypt one set of keys, and a key encrypting key pair shared with the ultimate recipient to key offset and encrypt the other set of keys.

The originator then sends the second set to the ultimate recipient.

(3) A Key Translation Center Environment

A Key Translation Center is used to translate keys for future communication between parties who:

(a) wish to communicate with each other but do not currently share keys,

(b) each share a key encrypting key pair with the Key Translation Center,

(c) have the ability (by the originator) to generate or otherwise acquire keys.

Both key encrypting keys and data keys may be translated and exchanged, though only one of the two types (key encrypting key or data key) is actually processed by the center at one time.

A key (key encrypting or data) to be used for future communication with the other party (the ultimate recipient) is sent to the Key Translation Center, encrypted under the offset key encrypting key pair shared between the originator and the Key Translation Center.

The Key Translation Center decrypts this key, reencrypts using notarization and the key encrypting key pair shared with the ultimate recipient, and sends the reencrypted version back to the originator.

The originator then sends the reencrypted version on to the ultimate recipient.

Multiple Key Translation Centers can be implemented as described in Appendix G.

6.3 Manual Key Distribution

Keys, all IVs (where needed), and accompanying documentation shall be protected throughout the distribution process.

The procedures shall ensure that:

(1) the key distribution is authorized,

(2) the key has been received by the authorized recipient,

(3) the key has not been disclosed, modified or replaced in transit.

Dual control with split knowledge shall be employed when keys are distributed in plaintext form.

Note: Requirements 1 and 2 could be fulfilled by the exchange and authentication of signature cards under separate cover. Requirement 3 could be fulfilled by the use, for example, of a bonded courier.

In Key Distribution Center (CKD) and Key Translation Center (CKT) environments, the key shared between a party and the center shall be a DEA key pair.

Originating and receiving parties shall identify to each other those individuals authorized to originate, receive and change keys and shall not reassign or delegate such responsibilities without proper notice.

These requirements shall also be implemented for distribution of IVs, where used.

Appendix B gives an example of a procedure for manual key distribution.

6.4 Automated Key Distribution

Automated key distribution is the electronic transmission of cryptographic keys (and, where needed, IVs) via a communication channel. Automated key distribution requires two types of keys:

(1) Key encrypting key(s) ((*)KK): shall only be used to encrypt and decrypt other keys (i.e., key encrypting keys and data keys), and shall never be used to encrypt, decrypt or authenticate data. Note that: (a) manual (*)KKs shall be initially distributed under dual control with split knowledge by some secure means before they can be used for the automated distribution of other keys; and (b) an asterisk is used to designate a key pair and an asterisk in parenthesis (*) means either a single key (KK) or a key pair (*KK).

(2) Data key(s) (KD): shall be used to encrypt and decrypt Initialization Vectors (IVs) (where desired) and to authenticate Cryptographic Service Messages. Note: data keys are also used to encrypt and decrypt or to authenticate data.

Since key management facility(s) can be designed to replace electronically distributed key encrypting keys and data keys automatically, manual intervention is kept to a minimum. Key encrypting keys generally have longer cryptoperiods than data keys.

Single key encrypting keys may be used by two parties communicating in a point-to-point environment (not a center environment). Key encrypting key pairs shall be used between a party and a center and may be used between communicating pairs in a point-to-point environment.

Two data keys with the same name may be sent in the same message.

For a keying relationship:

- When a new key (or keys) is (are) received, all stored keys of the same type (key encrypting keys or data keys) with the same name shall be replaced.

- When a key is discontinued, all keys of the same name (without regard to type) shall be discontinued.

Keys are sent along with other protected data to the intended recipient. The recipient shall be able to recognize the key so that it can be decrypted and loaded before any DEA process can begin. If the recipient has multiple key encrypting keys, information shall be sent identifying the key to be used. Keying material is automatically distributed using Cryptographic Service Messages.

7. Key and IV Encryption and Decryption

7.1 Notation

The following is the notation defined for this standard:

(1) Operators—are shown by the lower case letters:

a for authenticate
e for Electronic Code Book (ECB) encryption[1]
d for Electronic Code Book (ECB) decryption[1]

(2) ( ) indicates a choice of characters

(3) [ ] denotes field contents

(4) Concatenation—is indicated by "| |"

(5) Exclusive-or—is indicated by "+"

(6) Fields—are separated by "b"

(7) Subfields within a field—are separated by "."

(8) Field type and usage—is defined by a character sequence. The sequence for keys begins with a "K"; that for an IV begins with an "IV"; identifications begin with "ID". For keys, the second letter is defined by the table:

D for Data Key
K for Key Encrypting Key
N for Notarizing Key

For keys, the third letter is defined by the table:

o—the key has been key offset
l—the key is the left key of a key pair
r—the key is the right key of a key pair
U—the key has been notarized for the ultimate recipient
X—the data key is used in the computation of the error detection code

[1] ANSI X3.106-1983, Modes of Operation of the DEA.

otherwise, the third letter is unassigned and may be used for any purpose.

The types of keys used in this standard are:

KD	Data Key, plaintext or encrypted
KDU	Data Key, notarized for the ultimate recipient
KDX	Data Key, plaintext, for computation of the error detection code, hexadecimal 0123456789ABCDEF
KK	Key Encrypting Key, plaintext or encrypted
KKU	Key Encrypting Key, notarized for the ultimate recipient
KN	Notarizing Key

In this standard, IVs are denoted by:

IV	Initialization Vector, plaintext or encrypted

(9) Designation of Specific Keys and Key Pairs

DEA keys may be single keys or may be used as key pairs. A single key is expressed as:

$$K = key$$

whereas:

$$*KK = KKl \;||\; KKr$$

is a key pair consisting of two 64 bit quantities, part "l" and part "r". An asterisk "*" preceding the character sequence is used to specify that the key is a DEA key pair.

(*)KK designates the use of either a KK or *KK.

7.2 Encryption, Decryption, Authentication and Error Detection

7.2.1 General

Data keys shall be single DEA keys. Key encrypting keys may be a single DEA key or a DEA key pair. Key pairs should be used where additional security is needed (e.g., the data protected by the key(s) has a long security life).

A key pair shall not be encrypted or decrypted using a single key.

The general formats for encryption, decryption, and authentication and error detection, respectively are:

(encrypted quantity) = eK(plaintext quantity)

(decrypted quantity) = dK(encrypted quantity)

(authentication code) = MAC = aKD(data)

(error detection code) = EDC = aKDX(data) where KDX = 0123456789ABCDEF

The processes for encryption and decryption are specified below and in Figures II-IV.

7.2.2 Encryption and Decryption of a Single Key by a Single Key

Single KK and KD shall be encrypted and decrypted in the ECB mode using the following formulae:

KD encryption = eKKY(KD)
KD decryption = dKKY(KD)
KK encryption = eKKY(KK)
KK decryption = dKKY(KK)

7.2.3 Encryption and Decryption of a Single Key by a Key Pair

When KK and KD are encrypted by a *KK using DEA in the ECB mode, the following formulae shall be used:

Let the single key be "K( )Z" and the key pair be *KKY.

K( )Z encryption = ede*KKY(K( )Z) = eKKlY(dKKrY(eKKlY(K( )Z)))

K( )Z decryption = ded*KKY(K( )Z) = dKKlY(eKKrY(dKKlY(K( )Z)))

note that ( ) may be a "K" or a "D".

7.2.4 Encryption and Decryption of a Key Pair by a Key Pair

*KK pairs shall be encrypted and decrypted using the DEA in the ECB mode with key pairs using the following formulae:

Let *KKZ = KKlZ | | KKrZ

*KKY = KKlY | | KKrY

*KKZ encryption = ede*KKY(*KKZ) = eKKlY(dKKrY(eKKlY(KK1Z)))
| | eKKlY(dKKrY(eKKlY(KKrZ)))

*KKZ decryption = ded*KKY(*KKZ) = dKKlY(eKKrY(dKKlY(KK1Z)))
| | dKKlY(eKKrY(dKKlY(KKrZ)))

7.2.5 Implementation of Single Keys for Systems Using Key Encrypting Key Pairs

If KKlY = KKrY = KKY and K( )Z is a single key, then:

K( )z encryption = ede*KKY(K( )Z) = eKKY(K( )Z)
K( )Z decryption = ded*KKY(K( )Z) = dKKY(K( )Z)
note that ( ) may be a "K" or a "D".

The result is encryption (or decryption) of a single key by a single key. This allows facilities designed for use with key pairs to be downward compatible with those using a single key.

7.2.6 Encryption and Decryption of IVs

If desired, IVs may be encrypted and decrypted. IVs shall have a maximum length of 64 binary bits represented as 16 hexadecimal characters.

When IVs are encrypted and decrypted, they shall be encrypted and decrypted with the DEA in the ECB mode and using data keys in accordance with the following formulae:

encrypted IV = eKD(IV)
decrypted IV = dKD(encrypted IV)

7.2.7 Authentication of Cryptographic Service Messages

When Cryptographic Service Messages are to be authenticated, the MAC shall be computed using the technique defined in ANSI X9.9-1982, Section 4.0, using a data key. The input to the authentication algorithm shall be that obtained by editing according to section 4.3 of ANSI X9.9-1982.

MAC = aKD(data)

A Cryptographic Service Message containing a single KD shall be authenticated using that KD. If two KDs are sent in a Cryptographic Service Message, the key used for MAC computation shall be the exclusive-or of the two data keys.

7.2.8 Error Detection

When it is desired to detect transmission or process errors (i.e., when other means are not available), the authentication process of Section 7.2.7, above, shall be used with the fixed key, KDX = 0123456789ABCDEF to derive an error detection code (EDC).

EDC = aKDX(data)

7.3 Counters

7.3.1 Purpose

Counters are used to:

(a) detect Cryptographic Service Messages which have previously been used to transfer keys, and

(b) recognize a Cryptographic Service Message received out of sequence.

The exchange of an "expected count" permits parties to resynchronize counters that have otherwise become out of synchronization.

To achieve these objectives, counters shall only increment (never decrement) and shall never repeat during the use of the associated (*)KK.

Counters shall be independent of any counters that may be used in data messages.

Counters are transmitted as fields in selected Cryptographic Service Messages and are authenticated by an appended MAC. Error status messages are an exception in that they are transmitted without a MAC. Counter integrity may be provided by an appended error detection code (see Section 7.2.8).

Messages which use a manually distributed (*)KK to encrypt either KDs (two layer architecture) or automatically distributed keys (three layer architecture), shall use a counter which reflects the number of messages which have been transmitted using the particular manually distributed (*)KK for encryption. In a three layer architecture where only KD(s) encrypted under an automatically distributed (*)KK appear in a Cryptographic Service Message, the counter shall reflect the number of messages which have been transmitted using the particular (*)KK for encryption.

A (*)KK is often combined with a counter (offset) and the resulting key is used to encrypt another key for transmission (see Section 7.4, Offsetting of Keys, and Section 7.5, Notarization of Keys).

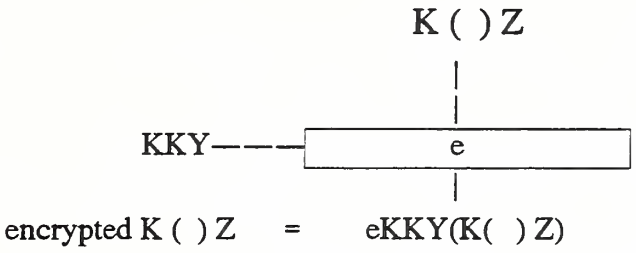
7.3.2 Counter Management

The contents of a counter are defined and shall be manipulated as a binary number. Counters shall always be set to one upon successfully loading the associated (*)KK, except in the case where a (*)KK is being sent in a Request For Service Message to a Key Translation Center. In that case alone, the counter associated with the (*)KK being sent to the center for translation shall be set to zero.

FIGURE II

ENCRYPTION/DECRYPTION OF A SINGLE KEY BY A SINGLE KEY

Encryption of a Single Key by a Single Key



Decryption of a Single Key by a Single Key

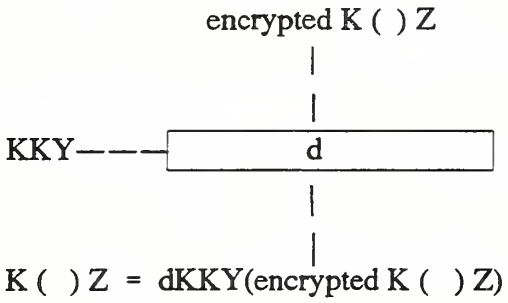
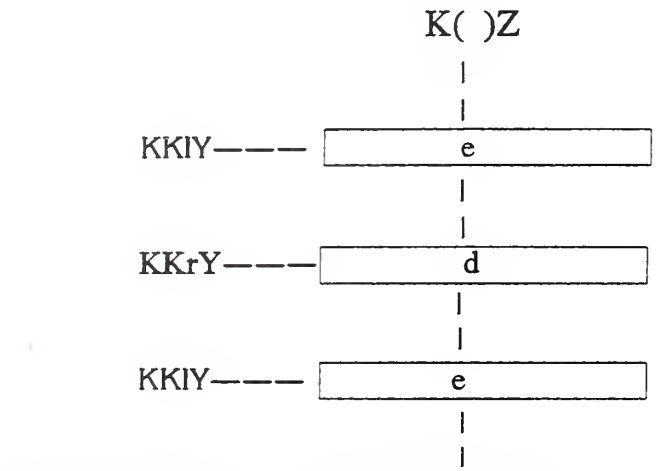


FIGURE III

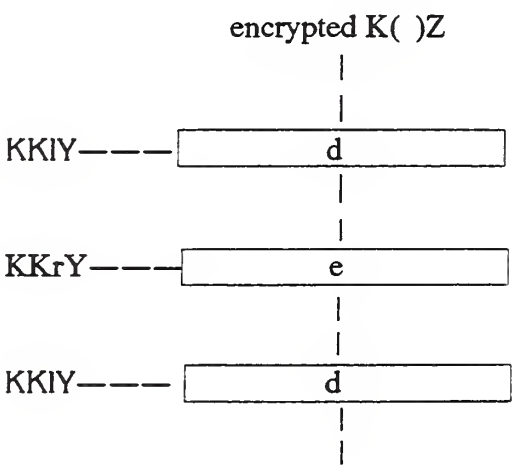
ENCRYPTION/DECRYPTION OF A SINGLE KEY BY A KEY PAIR

Encryption of a Single Key by a Key Pair



encrypted K( )Z = ede*KKY(K ( ) Z)

Decryption of a Single Key by a Key Pair

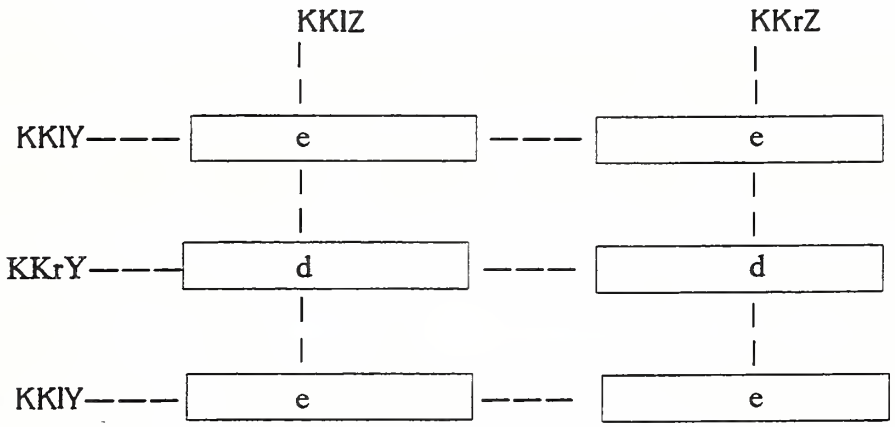


K ( ) Z = ded*KKY(encrypted K ( ) Z)

FIGURE IV

ENCRYPTION/DECRYPTION OF A KEY PAIR BY A KEY PAIR

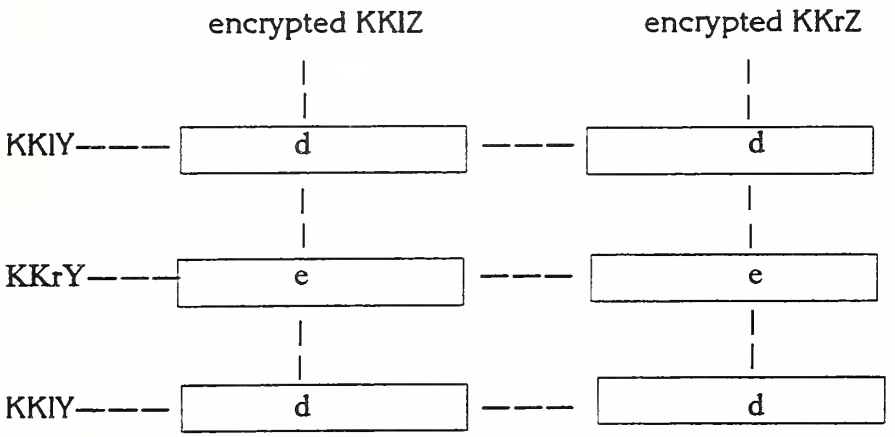
Encryption of a Key Pair by a Key Pair



encrypted *KKZ = ede*KKY(KKlZ) | | ede*KKY(KKrZ) =

encrypted KKlZ | | encrypted KKrZ

Decryption of a Key Pair by a Key Pair



*KKZ = ded*KKY(encrypted KKlZ) | | ded*KKY(encrypted KKrZ) =

KKlZ | | KKrZ

In point-to-point and Key Translation Center environments, two counters shall be maintained for each (*)KK shared between communicating pairs, an origination count and a reception count.

In a Key Distribution Center environment, keys are never sent to a Key Distribution Center; therefore, only one counter shall be maintained for each *KK shared between a party and a Key Distribution Center, i.e., the Key Distribution Center maintains an origination count while the various communicating parties maintain a (corresponding) reception count.

Under normal conditions, one party's origination count should equal the corresponding party's reception count.

Messages determined to be duplicates are not accepted and an error is reported to the originator. The recipient prepares an Error Service Message in which the nature of the error being reported (duplicate message), the value of the reception counter (the recipient's expected count) and the value of the origination count as received in the related message, are included.

A counter which resets to zero, is lost or lowered, shall be interpreted as a catastrophic error and shall invalidate the associated (*)KK. Detection of such a condition requires the enabling of an associated alarm. Recovery from such an event requires the use of a new (*)KK (either from storage or newly distributed) which shall initiate the value of the associated counters.

7.3.3 Cryptographic Service Message Management

When a recipient receives a Cryptographic Service Message whose count equals the expected (stored) count, the message is accepted. Both the originator's counter and recipient's reception counter should be incremented prior to the next message.

When the recipient receives a Cryptographic Service Message whose count is greater than the expected (stored) count, the message is accepted. The recipient's reception count is set to the received count plus one. This will be the new expected count.

When a recipient receives a Cryptographic Service Message whose count is less than the expected (stored) count, the message is not accepted and an error is reported to the originator. The recipient prepares an Error Service Message in which the nature of the error being reported (count error), the value of the reception counter (the recipient's expected (stored) count) and the value of the count as received in the related Cryptographic Service Message, are included.

The received count may be used to identify the Cryptographic Service Message which contained the counter in error. The party receiving the Error Service Message adjusts his origination counter up to the expected count value returned in the Error Service Message.

Alternatively, the party receiving the Error Service Message may elect to establish a new (*)KK with the recipient therein also initializing the value of the associated counters.

A summary of the above actions appears in Table I.

TABLE I
PROCESSING COUNTERS
(MAC's Check)

Received Count (CTA, CTB or CTP) equal to expected (stored) Count	Received Count (CTA, CTB, or CTP) greater than expected Count	Received Count (CTA, CTB, or CTP) less than expected (stored) Count	Action to be taken on receipt of an ESM or ERS	
			Received Count (CTA,CTB, or CTP) greater than Origination Count	Received Count (CTA, CTB, or CTP) Less Than or Equal to Origination (stored) Count
• Accept message • Generate RSM • Log (Optional) • Increment expected count (+1)	• Accept message • Generate RSM • Log (Mandatory) • Set expected count equal to received count +1	• Reject message • Send ESM with expected count • Log (Mandatory)	• Set origination (stored) count equal to expected count (CTA, CTB, or CTP) OR • Change (*)KK associated with the CTA, CTB, or CTP • Log (Mandatory) • Send new KSM or RTR with corrected count OR • Wait for new key request (RSI or RFS)	• Log (Mandatory) • Send New KSM or RTR with current origination (stored) count OR • Wait for new key request (RSI or RFS)

7.3.4 Other Considerations

In a Key Distribution Center or Key Translation Center environment, it is possible for a recipient to receive Cryptographic Service Messages whose counts are out of sequence, yet the Cryptographic Service Messages are authentic in that the MAC is correct. A party assuming the role of recipient may establish a window, representing a range of reception counter values, such that the corresponding Cryptographic Service Messages, should they arrive out of sequence, shall be accepted without declaring an error. A suggested method of defining and managing a window is included in Appendix F.

7.4 Offsetting of Keys

Offsetting of keys is the process of translating a (*)KK by exclusive-or'ing the key with a counter. Offsetting is always used to transform a (*)KK prior to encryption of a key by that (*)KK.

Let KB = (k1 k2, ... k7, p) be an eight bit key byte. The bit p may be set to give the byte odd parity. Let CB = (c1, c2..., c7, 0) be an eight bit byte formed from seven counter bits followed by a 0 bit. The exclusive-or of KB and CB is derived as follows:

$$KB + CB = (k1 + c1, k2 + c2, ...k7 + c7, p)$$

where p may be set to give the byte odd parity.

The exclusive-or of a 64 bit key, KK, with a 56-bit counter is derived by exclusive-or'ing the first byte of KK with the counter byte formed from the first seven high order bits of the counter (with a 0 bit added) to derive the first byte of the result. Then the second byte of KK is exclusive-or'ed with the counter byte formed from the second seven bits of the counter. The process continues until the eighth byte of KK is exclusive-or'ed with the counter byte formed from the last seven bits of the counter.

If the counter is represented as CT, the entire operation is indicated by the expression:

$$KKo = (KK + CT)$$

If *KK = KKl | | KKr is to be key offset by the counter CT, then the following equation is used:

$$*KKo = (KKl + CT) \mid\mid (KKr + CT)$$

7.5 Notarization of Keys

Notarization is a method for sealing keys with the identities of the communicating pair. Once sealed, or notarized, keys or IVs can only be recovered with knowledge of the key used to perform notarization and the identities of the communicating pair. A KD or a KK may be notarized before transmission by encryption using a notarizing key, (*)KN. (*)KN is formed by exclusive-or'ing (*)KK with a notary seal (NS).

When notarization is used the notarization field "NOS" shall be present, unless the use of notarization of a field is made explicit by the field tag (e.g., KKU, KDU).

In order to prevent ambiguities, the exclusive-or of a byte of key with either an ASCII character as defined in ANSI X3.4-1977, or a byte of DEA output shall be defined. Let:

KB = (k1, k2, ..., k7, p)

be an eight bit key byte. The bit p may be set to give the byte odd parity. Let:

AC = (p, b7, b6, ..., b1)

be an ASCII character. The exclusive-or of KB and AC is formed as follows:

KB + AC = (k1+b7, k2+b6, ..., k7+b1, p)

where p may be set to give the byte odd parity.

If OB = (01, 02,..., 08) is any DEA output byte, then

KB + OB = (k1+01, k2+02, ..., k7+07, p)

where p may be set to give the byte odd parity.

The exclusive-or of a DEA key and eight characters or bytes is the key formed by exclusive-or'ing each of the eight pairs of bytes.

The notarizing key, (*)KN, is formed from a (*)KK and the identities of the communicating pair. Suppose that Party A wishes to send a key to Party B. Let FM1 be the first eight characters of Party A's identity and let FM2 be the second eight characters of Party A's identity. Similarly let TO1 and TO2 represent the first and second halves of Party B's identity. If necessary, the identities are replicated to form sixteen character identifiers. For example, if Party A's identity is 'CITYB' and Party B's identity is 'MANHAN', then FM1 is 'CITYBCIT'; FM2 is 'YBCITYBC'; TO1 is 'MANHANMA'; and TO2 is 'NHANMANH'.

Case 1: Computing a Notarizing Key Using a KK

Let KK be the key which is to be used to compute the notarizing key. Then:

KKR = KK + FM1
KKL = KK + TO1
NS1 = eKKR(TO2)
NSr = eKKL(FM2)

NS = [(leftmost 32 bits of NS1) | | (rightmost 32 bits of NSr)] + CT( )

KN = KK + NS

KN is then used to encrypt (or notarize) either a KD or a KK.

Case 2: Computing a Notarizing Key Using a *KK

Let *KK be the key which is to be used to compute the notarizing key. Then:

*KK = KKl || KKr
KKR = KKr + FM1
KKL = KKl + TO1

Then:

NSl = eKKR(TO2) + CT ( )
NSr = eKKL(FM2) + CT ( )

Where CT ( ) is the counter used to key offset NSl and NSr, and the process for key offsetting is defined in section 7.4, above.

*KN = (KKl + NSl) || (KKr + NSr)

*KN is then used to encrypt (or notarize) either a KD or a (*)KK.

8. Cryptographic Service Messages

8.1 General

Cryptographic Service Messages shall be used for the automatic distribution and control of cryptographic keys and, where required, IVs, on a point-to-point basis and, optionally, with the support of a Key Distribution Center (CKD) or a Key Translation Center (CKT).

Centers may be used to reduce the number of manually distributed (*)KKs in large networks. A mutually trusted party is designated as the center. A *KK shared between any party and a center shall permit secure communications to be established between that party and any other party that has a *KK relationship with the center.

The Key Distribution Center has the capability to generate and send data keys for distribution. Key Distribution Centers may send keys unsolicited or upon request.

A Key Translation Center has the capability to transform and return keys for distribution by the originating party (e.g., a Key Translation Center does not require key generation capability; parties in the network have a peer relationship).

For audit and control purposes, Cryptographic Service Messages should be journalized.

8.2 Cryptographic Service Message Classes

The following Cryptographic Service Messages are defined by this standard:

(a) For Point-to-Point Key Management

(1) RSI Request Service Initiation Message

An optional message class that requests that a new keying relationship be initiated. The originator of an RSI might not have a key generation capability.

(2) KSM Key Service Message

A required message class that transfers a key from an originator to a recipient.

(3) RSM Response Service Message

A required message class that provides an authenticated response to a KSM.

(4) ESM Error Service Message

A required message class that reports an error in a previous Cryptographic Service Message.

(5) DSM Disconnect Service Message

An optional message class that is used to discontinue one or more keys.

(b) Additional Service Message Classes For Use With a Key Distribution Center (CKD) or Key Translation Center (CKT)

(1) RFS Request For Service Message

A required message class in a Key Translation Center environment that sends keys to a Key Translation Center to be translated for the ultimate recipient in the IDU field.

(2) RTR Response To Requestor Message

A required message class in a center environment that responds to an RFS, an ERS or an RSI to the center. An RTR may be initiated by a Key Distribution Center.

(3) ERS Error Recovery Service Message

A required message class that reports count and key errors to the Key Distribution Center or Key Translation Center and requests resynchronization of the count fields and re-initiation of service.

8.3 Cryptographic Service Message Character Set and Representation

The character set for Cryptographic Service Messages shall be the following characters: digits (0-9), letters (A-Z), comma (,) , period (.), space (b), solidus (/), hyphen (-), asterisk (*), open and close parentheses ((&)), carriage return and line feed. The characters (.) and (b) shall not be used in a subfield; (b) shall not be used in a field (except for the MAC field). The character (.) shall only be used in a field to separate subfields. All characters shall be represented as eight-bit characters (0, b7, ..., b1), where (b7, b6, ..., b1) are defined in ANSI X3.4-1977. Where

this necessitates a code translation, the translation shall be for internal processing and computational purposes only.

Hexadecimal characters shall be represented by the characters 0-9 and A-F.

8.4 Cryptographic Service Message Formats

(1) The presence of a Cryptographic Service Message is denoted by the financial message field tag, "CSM".

(2) A Cryptographic Service Message shall begin with an open parenthesis "(" and end with a close parenthesis ")".

(3) Field tags shall be separated from field contents by a solidus "/".

(4) Fields shall be separated by a blank "b" and, if desired for readability, a carriage return and line feed character(s).

(5) Subfields within a field shall be separated by a period "."

8.5 Cryptographic Service Message Fields and Subfields

Cryptographic Service Messages consist of a tag specifying that the message is a Cryptographic Service Message and a sequence of fields, subfields and associated parameters.

Fields containing keys may consist of up to four subfields.

(1) A key that has been encrypted or notarized shall always be the first subfield and is the only subfield required.

(2) The second subfield shall, if present, be a "P" to indicate that the plaintext key conforms to the specification for odd parity.

(3) The third subfield shall, if present, contain the identity of the key sent in the first subfield.

(4) The fourth subfield shall, if present, contain the identity of the key used to encrypt the key sent in the first subfield.

(5) Each subfield (whether present or not) shall be terminated by a period unless no subsequent subfield is present.

Examples of Key-Field Formats

(1) KD/key..IDK1, is a field with two subfields containing data. IDK1 is the identity assigned to the KD. The key used to encrypt the KD has not been explicitly defined. Hence its identity is assumed (is implicitly defined in the relationship).

(2) KK/key.P.IDK1.IDK2, is a field that contains a key, KK, that conforms to the specification for odd parity and subsequently has been encrypted under the key whose identity is IDK2; the identity of KK shall be IDK1.

(3) KDU/key, is a field with only one subfield that contains data. In this case, the identity of the new KD received in notarized form as KDU and the identity of the key used to decrypt KDU are implicitly defined.

(4) KD/key...IDK2, is a field with two subfields present. IDK2 is the identity of the key used to encrypt the KD. No name is assigned to the KD.

Unless otherwise determined by prior agreement, if two KDs are sent, the first KD shall be used by the ultimate recipient for authentication; the second shall be used for encryption.

8.6 Cryptographic Service Message Flow

8.6.1 General Cryptographic Service Message Flow

The fields and subfields shall be as defined in Table II. The fields used with each message class and their order shall be as defined in Tables III-V.

Normally, when a message is received with an option that is not implemented, an ESM with a "O" in the ERF field shall be returned. In the case of the EDC field, an ESM with an "O" in the ERF field may be returned or the field may be disregarded and processing may continue.

8.6.2 Point-to-Point Environment

Figures V and VI show the flow of Cryptographic Service Messages for the point-to-point environment. Table III defines the fields and their order for each message in this environment.

Message flow shall be as follows:

(1) If one party (Party B):

- wishes to communicate with another party (Party A),
- shares a (*)KK with Party A, and
- does not have key generation capability or access to keys,

then Party B sends a RSI to Party A requesting that Party A send key(s) and, optionally, an IV to Party B. If Party A receives a RSI from Party B with an error in it, an ESM shall be returned to Party B.

(2) If one party (Party A):

- wishes to send key(s) to another party (Party B),

- shares a (*)KK with Party B, and
- has key generation or acquisition capability,

then it generates or acquires keys and optionally an IV and sends a KSM to Party B containing the key(s) (and IV).

If a (*)KK is sent, it shall be encrypted under a (*)KK shared with Party B and the accompanying KD(s) sent in the Cryptographic Service Message shall be encrypted under the (*)KK sent in that message.

If a (*)KK is not sent, the KD(s) sent in the Cryptographic Service Message shall be encrypted under a (*)KK shared with Party B.

The IV (if encrypted) shall be encrypted under the KD (the second KD if two KDs are sent) in that Cryptographic Service Message.

The Cryptographic Service Message shall be authenticated using the KD(s) sent in that message.

Party B shall respond with a RSM if the KSM is received correctly, or with a ESM if there is an error in the received KSM. If Party A receives a RSM which contains an error(s), Party A shall return a ESM to Party B.

Party A may resend a KSM to Party B an arbitrary number of times, but Party A shall not send a new KSM (i.e., utilizing new keys or a new count for the (*)KK specified in a message) until the old KSM is acknowledged by a RSM or a ESM.

(3) If either party of a communicating pair wishes to terminate a keying relationship with the other party or wishes to discontinue the use of a specific key(s), that party may send a DSM to the second party.

The second party shall respond with a RSM if the DSM is received correctly and all information contained in the DSM was applicable. Otherwise, the second party shall respond with a ESM.

When a DSM is sent, the key named by the IDA field (or the only data key shared between the originating and recipient parties if no IDA field is present) shall be retained to authenticate the subsequent RSM. Note that that key shall then be discontinued. When a (*)KK is discontinued, all keys sent encrypted under that (*)KK shall also be discontinued without being named in the DSM.

When a RSM is received in error, no ESM shall be sent and manual recovery procedures are required.

8.6.3 Key Distribution Center (CKD) Environment

Figures VII and VIII show the flow of Cryptographic Service Messages for the Key Distribution Center environment when two parties each share a key encrypting key pair with the same Key Distribution Center. Table IV defines the fields and their order for each Cryptographic Service Message in this environment.

Cryptographic Service Message flow is as follows:

(1) If one party (Party B) wishes to communicate with another party (Party A), but does not share a key with Party A, Party B may use a Key Distribution Center. Party B must share a *KK with a Key Distribution Center that also has a *KK relationship with Party A. Party B may send a RSI to Party A requesting KD(s) and optionally an IV.

If Party A receives a RSI that contains errors, a ESM shall be returned to Party B.

(2) If Party A wishes to send KD(s) to another party (Party B), but does not share a (*)KK with Party B, Party A may use a Key Distribution Center. Party A must share a *KK with a Key Distribution Center that also shares a *KK with Party B. A RSI shall be sent to the Key Distribution Center requesting the KD(s) and optionally, an IV, for distribution to Party B, the ultimate recipient.

If the RSI received by the Key Distribution Center contains errors, a ESM shall be returned to Party A.

(3) If a Key Distribution Center receives a RSI from a party (Party A), it shall generate or acquire the requested KD(s) and optionally an IV. Optionally, a Key Distribution Center may generate a RTR without having received a RSI.

A RTR shall be sent to Party A containing two identical sets of KD(s). One set with the KD/ field tag shall be notarized using the *KK shared between Party A and the Center using the procedure of Section 7.5. The other set, with the KDU/ field tag shall be notarized using the *KK shared between Party B and the Center.

The IV, if encrypted, shall be encrypted under the KD (the second KD if two are present) sent in the Cryptographic Service Message. The Cryptographic Service Message shall be authenticated using the KD(s) contained in that message.

If the RTR received by Party A contains an error(s), Party A shall return a ESM to the center (CKD). When an unsolicited RTR is received by Party A, Party A shall respond with a RSM if no errors were detected.

(4) A KSM shall be sent by Party A to Party B containing the KDU(s) and the IV (if present) received in the RTR which caused the generation of the KSM. The KSM shall be authenticated using the KD(s) received in the RTR. If Party B receives the KSM from Party A without error, then Party B shall return a RSM to Party A. If the KSM received

by Party B contains errors, a ESM shall be returned to Party A. If the RSM received by Party A contains an error(s), then Party A shall respond to that RSM by sending a ESM back to Party B. A KSM may be re-sent until acknowledged by a RSM or a ESM.

(5) If Party A receives a ESM from Party B in response to a KSM identifying errors which can be attributed to Party B's relationship with the Key Distribution Center, then Party A shall send a ERS to the Key Distribution Center. These errors may be attributable to a CTB error, CKD unknown, IDK2 unknown and key parity errors in the received key. If a ERS received by the Key Distribution Center contains errors, then the Center shall reply to Party A with a ESM.

(6) If a Key Distribution Center receives a ERS, then another RTR shall be generated after the identified errors have been reconciled (see step 3, above).

(7) If either party of a communicating pair wishes to terminate a keying relationship with the other party or wishes to discontinue the use of a specific key(s), that party may send a DSM to the second party.

The second party shall respond with a RSM if the DSM is received correctly and all information contained in the DSM was applicable. Otherwise, the second party shall respond with a ESM.

When a DSM is sent, the key named by the IDA field (or the only data key shared between the originating and recipient parties if no IDA field is present) shall be retained to authenticate the subsequent RSM. Note that that key shall then be discontinued. When a (*)KK is discontinued, all keys sent encrypted under that (*)KK shall also be discontinued without being named in the DSM.

When a RSM is received in error, no ESM shall be sent and manual recovery procedures are required.

8.6.4 Key Translation Center (CKT) Environment

Figures IX and X show the flow of Cryptographic Service messages in a Key Translation Center environment when two parties each share key encrypting key pairs with the same Key Translation Center. Table V defines the fields and their order for each Cryptographic Service Message used in this environment.

Whenever a (*)KK relationship has been established between two parties in this environment, further exchange of keying material may be accomplished using procedures for the point-to-point environment. Cryptographic Service Message flow in the Key Translation Center environment shall be as follows:

(1) If one party (Party B) wishes to communicate with another party (Party A), but does not share a key with Party A, and does not have key generation capability, then Party

B may use a Key Translation Center. Party B must share a *KK with a Key Translation Center which also has a *KK relationship with Party A.

A RSI may be sent from Party B to Party A requesting the type(s) of key(s) and, optionally, an IV to be provided.

If the RSI received by Party A from Party B contains an error(s), then Party A shall return a ESM to Party B.

(2) If Party A wishes to send a key(s) to another party (Party but does not share a (*)KK with Party B, then Party A may use a Key Translation Center. Party A must have a key generation or acquisition capability and a *KK relationship with a Key Translation Center that also shares a *KK with Party B.

Party A shall send a RFS to the Key Translation Center, containing newly generated or acquired key(s) to be translated and eventually sent to Party B, the ultimate recipient.

If a (*)KK is sent, then one KD shall be sent. The (*)KK shall be encrypted under a *KK shared between Party A and the Key Translation Center; the KD shall be encrypted under the (*)KK sent in the message. If a (*)KK is not sent, then at least one and at most two KDs shall be sent, encrypted under a *KK shared between Party A and the Key Translation Center. The RFS shall be authenticated using the KD(s) sent in the message.

The Key Translation Center shall respond with a RTR if the RFS is received correctly, or with a ESM if there is an error in the received RFS. If Party A receives a RTR which contains an error(s), Party A shall return a ESM to the Key Translation Center.

(3) Upon receipt of a RFS by the Key Translation Center:

(a) The (*)KK, if present, shall be decrypted using the *KK shared between Party A and the Key Translation Center. The decrypted (*)KK shall be notarized using the *KK shared between the center and Party B, the count associated with the *KK and the identities of the two parties (A and B). It shall then be inserted as the (*)KKU field in a RTR for transmission to Party A. The KD in the RFS shall be used to authenticate both the RFS and the subsequent RTRs. It shall not be translated or inserted in the RTR.

(b) If a (*)KK is not present, then KD(s) shall be decrypted using the *KK shared between Party A and the center. The decrypted KD(s) shall be notarized using the *KK shared between the center and Party B, the count associated with the *KK and the identities of the two parties (A and B). They shall be inserted as the KDU field(s) in a RTR for transmission to Party A. The KD(s) shall be used to authenticate both the RFS and RTR.

If an error is detected in the RTR received by Party A, Party A shall return a ESM to the Key Translation Center.

(4) A KSM shall be sent by Party A to Party B containing the key(s) received in the RTR from the Key Translation Center. If a (*)KKU is present, KD(s) (generated or acquired) shall be inserted in the KSM in the KD field(s). The KD(s) shall be encrypted under the (*)KK sent to the Key Translation Center in the RFS that caused the RTR to be sent to Party A. KDU(S) are transferred directly from the RTR to the KSM.

If an IV (generated or acquired) is to be encrypted, it shall be encrypted under the KD (the second if two KDs are present) sent in the KSM.

The KSM shall be authenticated using the KD(s) sent in the message.

Party A may resend a KSM to Party B an arbitrary number of times, but Party A shall not send a new KSM (i.e., utilizing new keys or a new count for the (*)KK specified in a message) until the old KSM is acknowledged by a RSM or a ESM.

If an error is detected by Party B in the KSM sent by Party A, then a ESM shall be returned to Party A. Otherwise, a RSM shall be returned to Party A. If Party A detects an error in the RSM, Party A shall return a ESM to Party B in response to the RSM.

(5) If a ESM is received by Party A from Party B in response to a KSM with errors attributable to the relationship between the Key Translation Center and Party B, then a ERS shall be sent to the Key Translation Center. New keys shall be generated or acquired.

If a ERS received by the Key Translation Center contains errors, then the center shall reply to Party A with a ESM.

(6) Upon receipt of a ERS from Party A, the Key Translation Center shall prepare another RTR (see step 3, above) after resolving any problems in the Key Translation Center-to-Party B (ultimate recipient) relationship.

(7) If either party of a communicating pair wishes to terminate a keying relationship with the other party or wishes to discontinue the use of a specific key(s), that party may send a DSM to the second party.

The second party shall respond with a RSM if the DSM is received correctly and all information contained in the DSM was applicable. Otherwise, the second party shall respond with a ESM.

When a DSM is sent, the key named by the IDA field (or the only data key shared between the originating and recipient parties if no IDA field is present) shall be retained to authenticate the subsequent RSM. Note that that key shall then be discontinued.

When a (*)KK is discontinued, all keys sent encrypted under that (*)KK shall also be discontinued without being named in the DSM.

When a RSM is received in error, no ESM shall be sent and manual recovery procedures are required.

TABLE II

SERVICE MESSAGE FIELDS AND SUBFIELDS

ACRONYM	NAME	DEFINITION/REMARKS	SPECIFICATION
CTA	Count, "A"	An incrementing binary counter. Associated with an *KK used to encrypt either an (*)KK or KD(s) sent in a Cryptographic Service Message. Set to one upon installation of this new *KK. Used between a CKD or a CKT and another party designated as "A".	Up to 14 hex characters; leading zeros may be suppressed for transmission.
CTB	Count, "B"	An incrementing binary counter. Associated with an *KK used to encrypt either an (*)KKU or KDU(s) sent in a Cryptographic Service Message. Set to one upon installation of this new *KK. Used between a CKD or a CKT and another party designated as "B".	Up to 14 hex characters; leading zeros may be suppressed for transmission.
CTP	Count, Pair	An incrementing binary counter. Associated with an (*)KK or KD(s) sent in a Cryptographic Service Message. Set to one upon installation of this new (*)KK. Used between communicating pairs, but not between a CKD or a CKT and another party.	Up to 14 hex characters; leading zeros may be suppressed for transmission.
CTR	Count, "R"	A count field of an error message which is equal to the received count and is sent only when a count error occurs.	Up to 14 hex characters; leading zeros may be suppressed for transmission.

TABLE II

SERVICE MESSAGE FIELDS AND SUBFIELDS (Continued)

ACRONYM	NAME	DEFINITION/REMARKS	SPECIFICATION
EDC	Error Detection Code	The Error Detection Code when used shall be generated on all components of the associated service message (ERS for the CKD environment, ESM or RSI) using the editing, computation and formating requirements for a MAC of ANSI X9.9-1982. The hexadecimal key for EDC computation shall be 0123456789ABCDEF.	9 characters (4 hex) b (4 hex)
EDK	Effective Date of Key	Date and Coordinated Universal Time of KD activation.	12 characters YYMMDDHHMMSS
ERF	Error Field	See Error Codes and Definitions, below	Up to 16 characters.

Error Code	Definition
A	CTA error
B	CTB error
C	Cannot process[1]
D	CKD unknown
E	Facility inoperative
F	Format (syntax) error
G	Reserved
H	User defined
I	Key identifier not known to recipient
K	Parity error in received key
M	MAC error (failure to authenticate)

[1] Optional. May be used as a general error code where a more specific error code is not appropriate.

TABLE II

SERVICE MESSAGE FIELDS AND SUBFIELDS (Continued)

ACRONYM	NAME	DEFINITION/REMARKS	SPECIFICATION
ERF	Error Field	(Continued)	
	O	Option not implemented	
	P	CTP error	
	T	CKT unknown	
	U	IDU not known to the CKT or CKD	
	X	EDC error (probable transmission error)	
IDA	Authentication (MAC) Key	Identifies the key to be used to authenticate a DSM. This key shall be discontinued.	Up to 16 characters
IDC	Identity of CKD or CKT	Identity of CKD or CKT to be used.	Between 4 and 16 characters, inclusive
IDD	Discontinued Key	Identifies key to be discontinued.	Up to 16 characters
IDK1	Key Identifier	Identifies (names) the received key.	Up to 16 characters (subfield)
IDK2	Key Encrypting Key Identifier	Identifies (names) the (*)KK used to encrypt/decrypt a (*)KK or a KD.	Up to 16 characters (subfield)
IDU	Identity of Ultimate Recipient	This field is only used with a CKT or a CKD.	4 to 16 characters.

TABLE II

SERVICE MESSAGE FIELDS AND SUBFIELDS (Continued)

ACRONYM	NAME	DEFINITION/REMARKS	SPECIFICATION
IV	Initialization Vector	Starting point for DEA encryption/decryption process. If encrypted, is always encrypted under a data key (KD). If only one data key is transmitted in a message, it shall be encrypted under that KD. If two KDs are transmitted, the IV shall be encrypted under the second KD. If the first character is an E, the IV shall be encrypted. If it is a P, the IV shall be sent in plain text form.	1 character followed by up to 16 hex characters. Leading zeros of an encrypted IV shall not be suppressed for transmission.
KD	Data Key	Encrypted KD. May be used for encryption or authentication. A maximum of two KD fields can be sent per message. If two KDS are sent, the first shall be used for authentication and the second shall be used for encryption. In a point-to-point environment, the KD(s) sent in this field shall be notarized if and only if no KK field is present and the NOS field is present.	16 hex characters
KDU	Data Encrypting Key, Notarized	A KD encrypted under the *KN generated by the CKT or CKD for the ultimate recipient specified in the IDU field of an RTR or specified by the RCV field of a KSM that retransmits this key to the ultimate recipient. Up to two fields can be sent per message. If two KDUs are sent, the first shall be used for authentication and the second shall be used for encryption.	16 hex characters

TABLE II

SERVICE MESSAGE FIELDS AND SUBFIELDS (Continued)

ACRONYM	NAME	DEFINITION/REMARKS	SPECIFICATION
KK	Key Encrypting Key	Encrypted KK. In a point-to-point environment, the KK sent in this field shall be notarized if and only if the NOS field is present.	16 hex characters
*KK	Key Encrypting Key Pair	Encrypted KK Pair. In a point-to-point environment, the *KK sent in this field shall be notarized if and only if the NOS field is present.	32 hex characters
KKU	Key Encrypting Key, Notarized	A KK encrypted under the *KN generated by the CKT or CKD for the ultimate recipient, as specified in the IDU field.	16 hex characters
*KKU	Key Encrypting Key Pair Notarized	A KK pair encrypted under the *KN generated by the CKT or CKD for the ultimate recipient, as specified in the IDU field.	32 hex characters
MAC	Message Authentication Code	The MAC shall be generated on all components of the associated Cryptographic Service Message using the editing, computation and format requirements of ANSI X9.9-1982.	9 characters (4 hex) b (4 hex)
MCL	Message Class	Type of Cryptographic Service Message.	3 Characters; DSM, ERS, ESM, KSM, RFS, RSI, RSM or RTR
NOS	Notarization Indicator	Indicates use of notarization process for (*)KK if (*)KK is present in a message. If no (*)KK is present, the KD(s) is (are) notarized.	Zero length field
ORG	Originator	Cryptographic Service Message originator.	Between 4 and 16 characters, inclusive

TABLE II

SERVICE MESSAGE FIELDS AND SUBFIELDS (Continued)

ACRONYM	NAME	DEFINITION/REMARKS	SPECIFICATION
P	Key Parity	Used to indicate that the plaintext key conforms to the specification for odd parity.	1 character (subfield)
RCV	Recipient	Cryptographic Service Message recipient.	Between 4 and 16 characters, inclusive
SVR	Service Request	Specifies type of service requested. One data key is implicitly requested. KD requests two data keys. KK requests a key encrypting key. An *KK requests a key encrypting key pair (a KK or a *KK may be requested, but not both). IV requests an IV. The IV shall be encrypted unless, by prior agreement, a plaintext IV is to be sent. A minimum of one and a maximum of three keys and an IV may be requested in a single ERS or RSI. Types requested shall be separated by periods.	0 to 9 characters

TABLE III

FIELDS USED WITH EACH MESSAGE CLASS:

POINT-TO-POINT ENVIRONMENT

The fields which shall be used with each message class and the order in which they shall appear are specified below:

RSI[1]	KSM	RSM (Responding to KSM)	ESM (Responding to KSM)	DSM[1]
MCL	MCL	MCL	MCL	MCL
RCV	RCV	RCV	RCV	RCV
ORG	ORG	ORG	ORG	ORG
SVR	NOS[3]	MAC	CTP	IDD[8]
EDC[1]	(*)KK[1]		CTR[7]	IDA[1]
	KD[2]		ERF	MAC
	IV[1]		EDC[1]	
	EDK[1]			
	CTP			
	MAC			

RSM (Responding to DSM)	ESM (Responding to DSM, RSI or RSM)
MCL	MCL
RCV	RCV
ORG	ORG
IDD[9]	ERF
MAC	EDC[1]

Explanations:

1 Optional

2 A maximum of two such fields may be sent in a message.

3 Required when notarization of keys is used.

4 Present if and only if there is a (*)KKU field.

5 Present if and only if there is no (*)KKU field.

6 Only one KD field shall be present if a (*)KK field is present.

7 Required if and only if a count error occurs.

8 Any number of such fields may be sent in a DSM. Each IDD field shall either contain the identity of a discontinued key or shall be a null field. A null field indicates that the keying relationship shall be discontinued. See Section 10.2 (4).

9 The number of IDD fields shall be equal to the number of IDD fields in the DSM to which this RSM responds. Each IDD field shall either contain the identity of a discontinued key or shall be a null field. A null field indicates that the keying relationship shall be discontinued. See Section 10.8 (5).

TABLE IV

FIELDS USED WITH EACH MESSAGE CLASS: KEY DISTRIBUTION CENTER ENVIRONMENT

The fields which shall be used with each message class and the order in which they shall appear are specified below:

RSI[1] (to CKD)	RSI[1] (from B to A)	ESM (Responding to RSI to CKD or ERS)	RTR	RSM (Responding to an unsolicited RTR)
MCL	MCL	MCL	MCL	MCL
RCV	RCV	RCV	RCV	RCV
ORG	ORG	ORG	ORG	ORG
IDU	IDC	IDU	IDU	IDU
SVR	SVR	ERF	KD[2]	MAC
EDC[1]	EDC[1]	EDC[1]	KDU[2]	
			IV[1]	
			EDK[1]	
			CTB	
			CTA	
			MAC	

ESM (Responding to RTR)	ESM (Responding to RSI from B to A)	KSM	RSM (Responding to KSM)	ESM (Responding to KSM)
MCL	MCL	MCL	MCL	MCL
RCV	RCV	RCV	RCV	RCV
ORG	ORG	ORG	ORG	ORG
IDU	IDC	IDC	IDC	IDC
CTA	ERF	KDU[2]	MAC	CTB
CTR[7]	EDC[1]	IV[1]		CTR[7]
ERF		EDK[1]		ERF
EDC[1]		CTB		EDC[1]
		MAC		

TABLE IV (Continued)

ERS	ESM (Responding to RSM sent in response to a KSM)	DSM[1]	RSM(Responding to DSM)	ESM(Responding to DSM)
MCL	MCL	MCL	MCL	MCL
RCV	RCV	RCV	RCV	RCV
ORG	ORG	ORG	ORG	ORG
IDU	IDC	IDD[8]	IDD[9]	ERF
ERF	ERF	IDA[1]	MAC	EDC1
SVR	EDC[1]	MAC		
CTB				
CTR[7]				
EDC[1]				

NOTES:

1 Optional
2 A maximum of two such fields may be sent in a message.
3 Required when notarization of keys is used.
4 Present if and only if there is a (*)KKU field.
5 Present if and only if there is no (*)KKU field.
6 Only one KD field shall be present if a (*)KK field is present.
7 Required if and only if a count error occurs.
8 Any number of such fields may be sent in a DSM. Each IDD field shall either contain the identity of a discontinued key or shall be a null field. A null field indicates that the keying relationship shall be discontinued. See Section 10.2 (4).
9 The number of IDD fields shall be equal to the number of IDD fields in the DSM to which this RSM responds. Each IDD field shall either contain the identity of a discontinued key or shall be a null field. A null field indicates that the keying relationship shall be discontinued. See Section 10.8 (5).

TABLE V

FIELDS USED WITH EACH MESSAGE CLASS: KEY TRANSLATION CENTER ENVIRONMENT

The fields which shall be used with each message class and the order in which they shall appear are specified below:

RSI[1]	ESM (Responding to RSI)	RFS	ESM (Responding to ERS, RFS)
MCL	MCL	MCL	MCL
RCV	RCV	RCV	RCV
ORG	ORG	ORG	ORG
IDC	IDC	IDU	IDU
SVR	ERF	(*) KK[1]	CTA
EDC[1]	EDC[1]	KD[2,6]	CTR[7]
		CTA	ERF
		MAC	EDC[1]

RTR(Responding to ERS or RFS)	ESM(Responding to RTR)	KSM	RSM(Responding to KSM)	ESM(Responding to KSM)
MCL	MCL	MCL	MCL	MCL
RCV	RCV	RCV	RCV	RCV
ORG	ORG	ORG	ORG	ORG
IDU	IDU	IDC	IDC	IDC
(*)KKU[1]	ERF	(*)KKU[1]	MAC	CTB
KDU[2,5]	EDC[1]	KD[2,4]		CTR[7]
CTB		KDU[2,5]		ERF
MAC		IV[1]		EDC[1]
		EDK[1]		
		CTB		
		MAC		

TABLE V (Continued)

ERS	ESM (Responding to RSM sent in response to a KSM)	DSM1	RSM(Responding to DSM)	ESM(Responding to DSM)
MCL	MCL	MCL	MCL	MCL
RCV	RCV	RCV	RCV	RCV
ORG	ORG	ORG	ORG	ORG
IDU	IDC	IDD[8]	IDD9	ERF
(*) KK1	ERF	IDA	MAC	EDC[1]
KD[2,6]	EDC1	MAC		
ERF				
CTB				
CTR[7]				
CTA				
MAC				

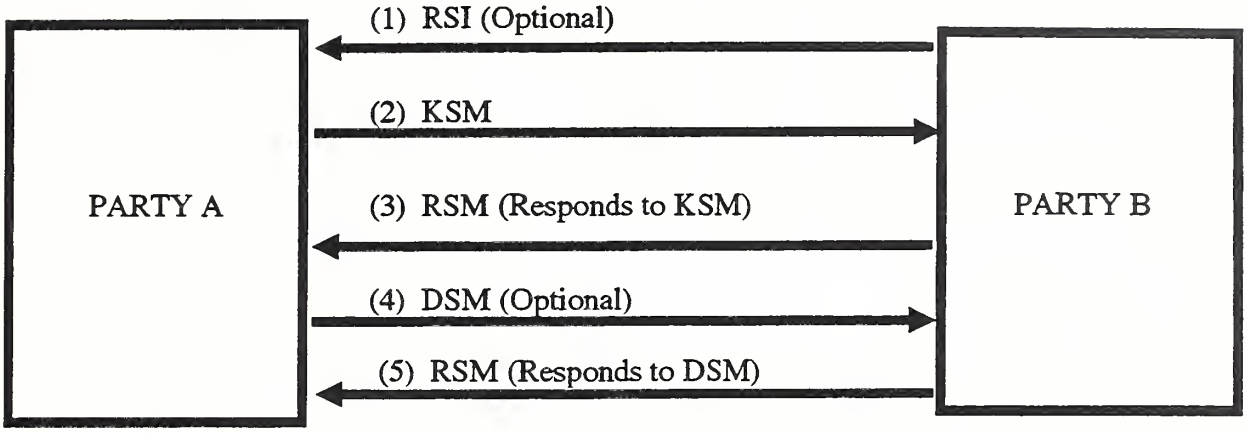
NOTES:

1 Optional
2 A maximum of two such fields may be sent in a message.
3 Required when notarization of keys is used.
4 Present if and only if there is a (*)KKU field.
5 Present if and only if there is no (*)KKU field.
6 Only one KD field shall be present if a (*)KK field is present.
7 Required if and only if a count error occurs.
8 Any number of such fields may be sent in a DSM. Each IDD field shall either contain the identity of a discontinued key or shall be a null field. A null field indicates that the keying relationship shall be discontinued. See Section 10.2 (4).
9 The number of IDD fields shall be equal to the number of IDD fields in the DSM to which this RSM responds. Each IDD field shall either contain the identity of a discontinued key or shall be a null field. A null field indicates that the keying relationship shall be discontinued. See Section 10.8 (5).

FIGURE V

POINT-TO-POINT ENVIRONMENT

(Normal Message Flow)

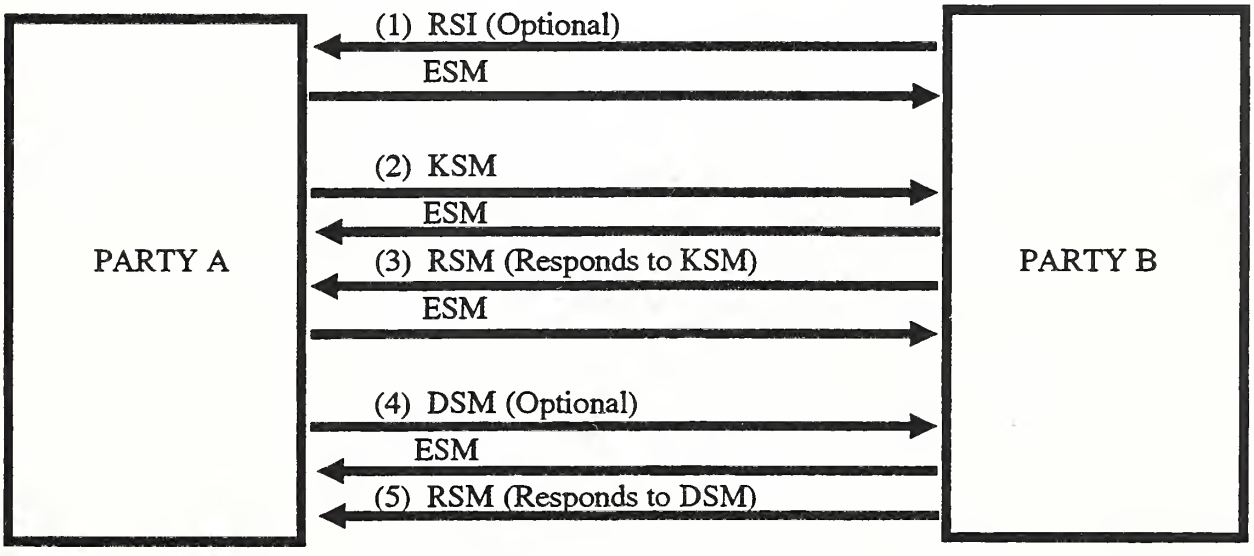


Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

FIGURE VI

POINT-TO-POINT ENVIRONMENT

(Message Flow With Error Messages)

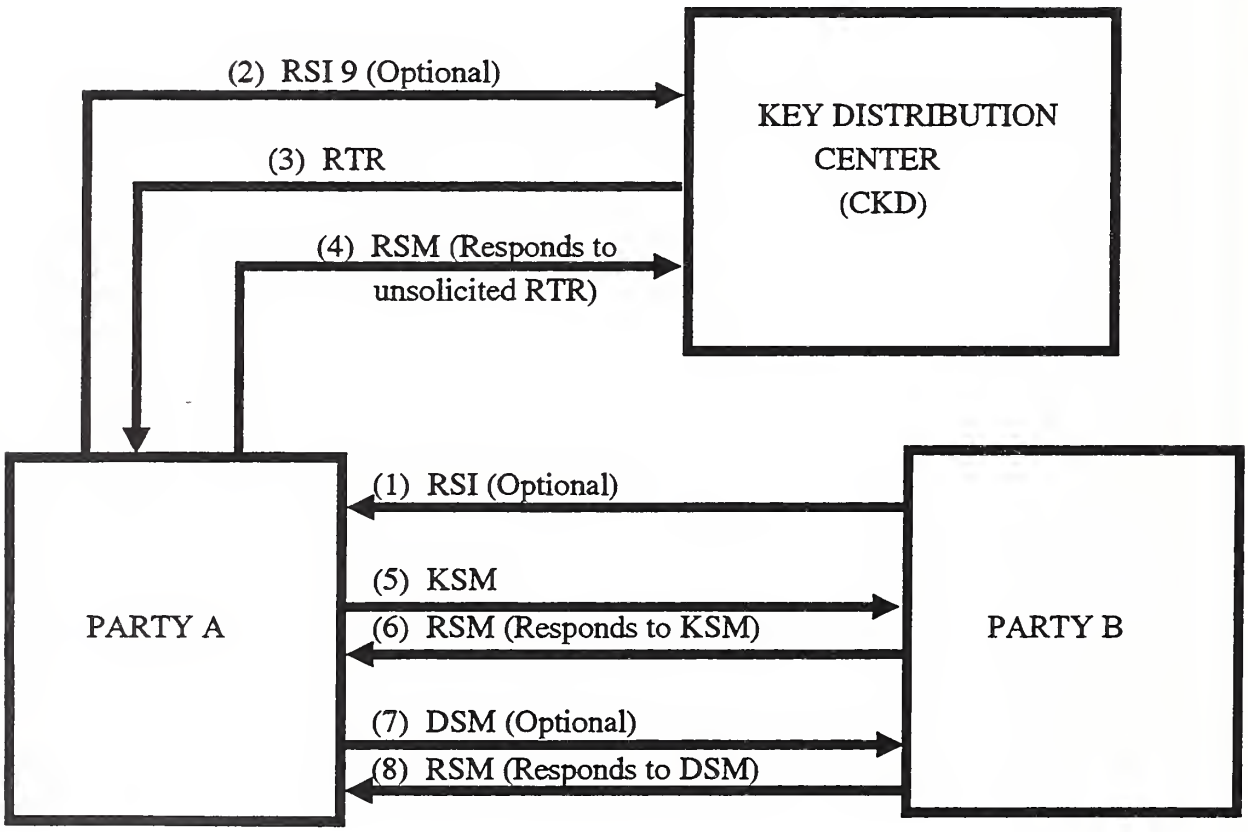


Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

FIGURE VII

KEY DISTRIBUTION CENTER ENVIRONMENT

(Normal Message Flow)

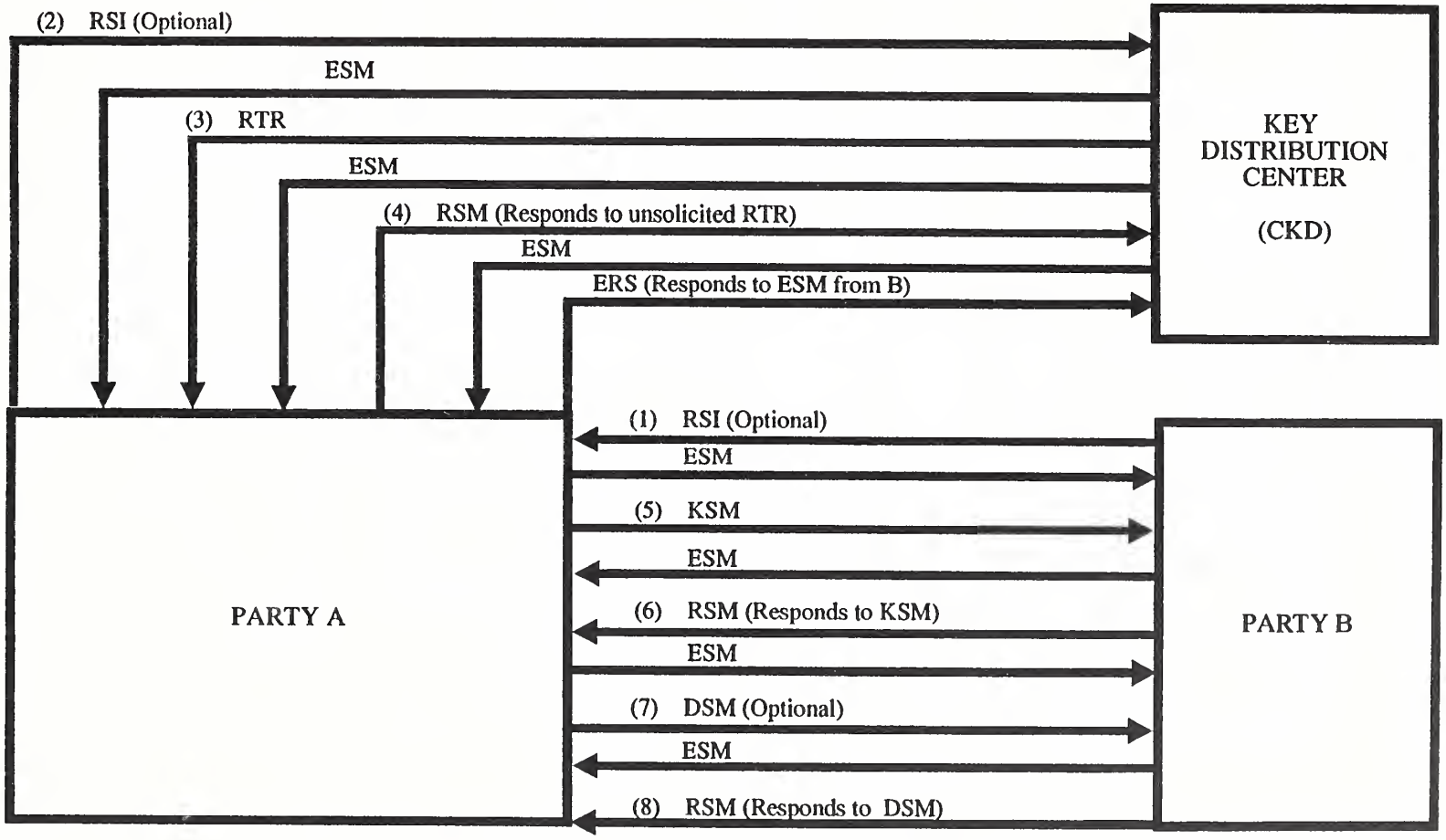


Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

FIGURE VIII

KEY DISTRIBUTION CENTER ENVIRONMENT

(Message Flow With Error Messages)

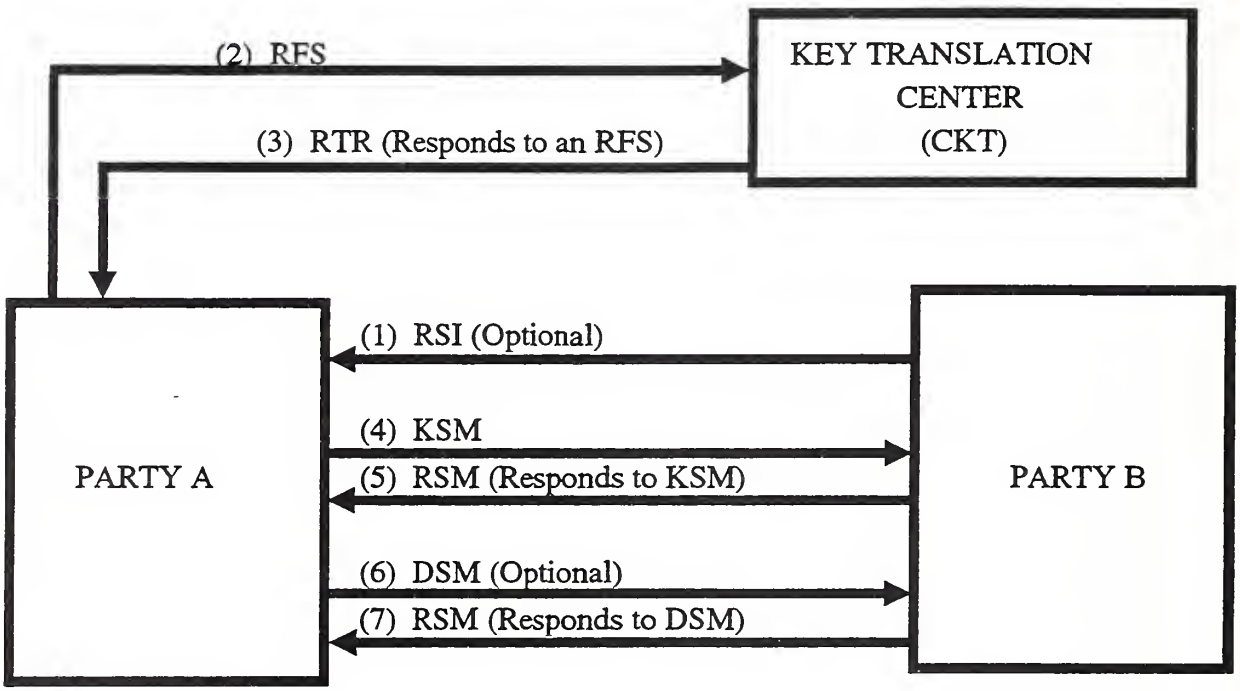


Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

FIGURE IX

KEY TRANSLATION CENTER ENVIRONMENT

(Normal Message Flow)

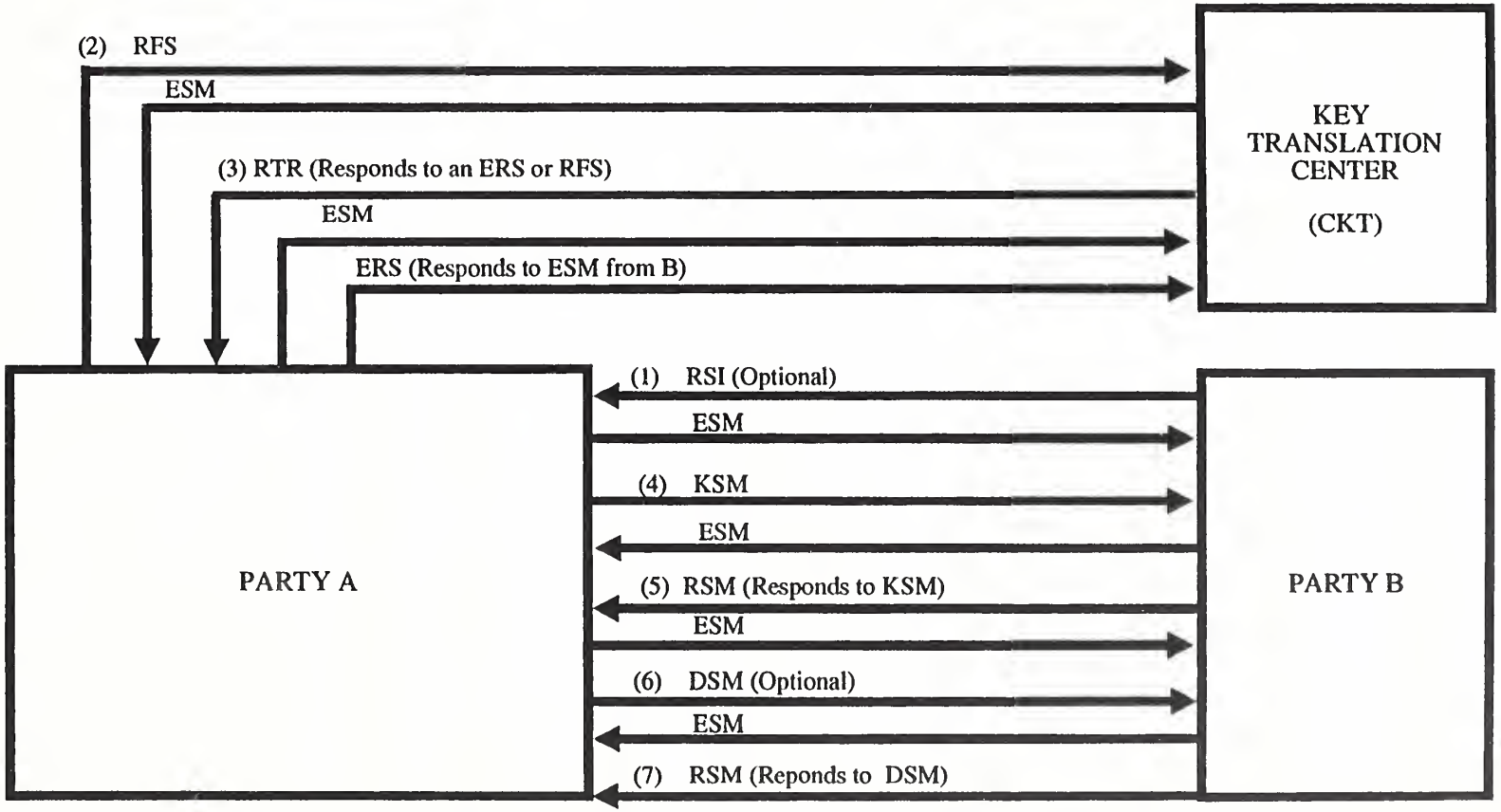


Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

FIGURE X

KEY TRANSLATION CENTER ENVIRONMENT

(Message Flow With Error Messages)



Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

9. Generating Cryptographic Service Messages

9.1 Cryptographic Service Message Class Determination

The class of the outgoing Cryptographic Service Message is specified by the first field (the three letters following "CSM(MCL/"). The following table references the sections of this standard which shall be used in generating each class of Cryptographic Service Message.

If the MCL Field Content Is:	The Cryptographic Service Message Shall Be Generated in Accordance With:	Page Number:
DSM	Section 9.2	56
ERS	Section 9.3	57
ESM	Section 9.4	61
KSM	Section 9.5	64
RFS	Section 9.6	70
RSI	Section 9.7	73
RSM	Section 9.8	74
RTR	Section 9.9	76

9.2 Generating A Disconnect Service Message

A Disconnect Service Message (DSM) is generated in order to discontinue one or more keys or to terminate a keying relationship. It may be sent by either party of the relationship. Disconnect Service Messages shall be generated by computing or selecting field contents in accordance with the following process.

Process	Field	Action
1.	MCL	Insert DSM in the field. The field becomes: MCL/DSM
2.	RCV	Insert recipient's identity in the field. If RRRR is the identity, the field becomes: RCV/RRRR
3.	ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO

Process	Field	Action
4.	IDD	Insert the identity of the key to be discontinued. Use a separate IDD field for each such key to be discontinued. If a keying relationship is to be terminated, the IDD field shall be null.
5.	IDA	(Not required if the originating and recipient parties share one and only one data key) Insert the identity of the key to be used to authenticate this DSM. Note that this key shall also be named in an IDD field (unless the IDD field is null and the keying relationship is to be discontinued).
6.	MAC	The MAC field contents shall be computed using the KD as follows: aKD(MCL/DSMb ... bIDA/IDK1b) and if brackets are used to denote field contents, the field becomes: MAC/[aKD(MCL/DSMb ... bIDA/IDK1b)] When the DSM has been generated, the key(s) named in the IDD field(s) may be discontinued (excluding the key used to authenticate this message, which shall be retained to authenticate the RSM responding to this DSM).

9.3 Generating An Error Recovery Service Message

An Error Recovery Service Message (ERS) is sent to a CKD or CKT by the originator of a KSM (i.e., Party A) in response to an ESM received from the recipient of the KSM (i.e., Party B) (see Figures VIII and X). This Cryptographic Service Message is used to

(1) announce to the CKD or CKT that errors were received in the KSM (by Party B) which are attributable to a problem in the key or count shared by Party B and the CKD or CKT, and

(2) request that further keys be processed (after appropriate corrections have been made) to be sent by Party A to Party B.

Error Recovery Service Messages shall be generated by computing or selecting field contents in accordance with the following process.

Process	Field	Action
1.	MCL	Insert ERS in the field. The field becomes: MCL/ERS
2.	RCV	Insert identity of the CKD or CKT in the field. If RRRR is the identity, the field becomes: RCV/RRRR
3.	ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
4	IDU	Insert the identity of the ultimate recipient in the field. If UUUU is the identity, the field becomes: IDU/UUUU
5.	(*)KK	(CKT environment only) (Optional) In using an ERS to recover, new key(s) shall be generated or acquired.

Optional Subfields

If it is desired to use the odd parity feature, to name the (*)KK being sent in the Cryptographic Service Message, to specify the key encrypting key to be used to decrypt the (*)KK (or any combination thereof), form and insert the applicable subfields using the rules of section 8.5.

Let (*)KKZ be the key to be encrypted (i.e., to be sent to the IDU party) and *KKY be the key encrypting key shared with the CKT.

Use the method of section 7.4 to compute the offset, *KKoY.

The encrypted (*)KK is computed using:

encrypted KKZ = ede*KKoY(KKZ) or

encrypted *KKZ = ede*KKoY(*KKZ)

If brackets are used to denote the field contents, then the field becomes:

KK/[ede*KKoY(KKZ) (optional subfields)] or

*KK/[ede*KKoY(*KKZ) (optional subfields)]

Process	Field	Action
6.	KD	(CKT environment only)
		In using an ERS to recover, new keys shall be generated or acquired.
	Optional Subfields	
		If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message or to specify the key encrypting key to be used to decrypt the KDs (or any combination thereof), form and insert the applicable subfields using the rules of Section 8.5.
		Case 1: The ERS contains a (*)KK field
		One and only one KD shall be sent in the ERS in the KD field. That KD shall be used to authenticate the ERS and to generate the MAC for inclusion in the RTR that responds to this ERS.
		Let KDI be the key to be sent in this field. Let (*)KKY be the key encrypting key sent in the message. The KD is computed using the equations:
		The KD is encrypted by a (*)KK; an offset of zero is used.
	(a)	Encrypt the KDI using the equation:
		encrypted KDI = eKKoY(KDI) or
		encrypted KDI = ede*KKoY(KDI)
	(b)	If brackets are used to denote field contents, the field becomes:
		KD/[eKKoY(KDI) .P] or
		KD/[ede*KKoY(KDI) .P]
		where P is an optional subfield.
		Case 2: There is no (*)KK field in the message.
		At least one and at most two KDs shall be sent in a ERS as KD field(s).
		A KD shall be encrypted by a *KK, and the following equations are used:
	(a)	Use the procedure of Section 7.4 to compute the *KKoY.
	(b)	Encrypt the KDI using the equation:
		encrypted KDI = ede*KKoY(KDI)
	(c)	If brackets are used to denote field contents, the field becomes:

Process	Field	Action
		KD/[ede*KKoY(KDI) (optional subfields)]
7.	ERF	Copy error codes applicable to the Party B/Center relationship from the ERF field received in the ESM from which the ERS is generated, to the ERF field.
8.	SVR	(CKD environment only) Insert the subfields designating the type of service requested. Note that a single data key is implicitly requested by the presence of a SVR field, and that an (*)KK shall not be requested. E.g., SVR/KD to request two data keys SVR/KD.IV to request two data keys and an encrypted IV
9.	CTB	Let "b" be the contents of the CTB field received in the ESM from which the ERS is generated. Then the contents of the CTB field in the ERS are set equal to "b" and the field becomes: CTB/b
10.	CTR	(used when a count error has been detected) Let "r" be the contents of the CTR field received in the ESM from which the ERS is generated. Then the contents of the CTR field in the ERS are set equal to "r" and the field becomes: CTR/r
11.	CTA	(CKT environment only) If the value of the CTA before ERS is "a" then the ERS shall contain the CTA field: CTA/a
12.	MAC	(CKT environment only) The MAC is always computed using the KDs sent in the message. If only one KD is sent, KDJ, then that key shall be used. If two KDs are sent, KDH and KDI, then the key, KDJ (used to authenticate the Cryptographic Service Message), is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ(MCL/ERSb...bCTA/ab)

Process	Field	Action
		and if brackets are used to denote the field contents, the field becomes: MAC/[aKDJ(MCL/ERSb...bCTA/ab)] Input to the authentication algorithm starts with the first character following the open parenthesis "(" of the Cryptographic Service Message, and continues through the blank, "b", preceding the MAC field.
13.	EDC	(CKD environment only) (Optional) The data key for EDC computation shall be: KDX = 0123456789ABCDEF The EDC is computed using: EDC = aKDX(MCL/ERSb ... b) and if brackets are used to denote field contents, the field becomes: EDC/[aKDX(MCL/ERSb...b)] Input to the authentication algorithm starts with the first character following the open parenthesis "(" of the Cryptographic Service Message, and continues through the blank, "b", preceding the EDC field.

9.4 Generating An Error Service Message

An Error Service Message (ESM) is sent in response to the detection of one or more of the following error conditions in a Cryptographic Service Message (other than an ESM):

Error Code	Definition
A	CTA error
B	CTB error
C	Cannot process
D	CKD unknown
E	Facility inoperative
F	Format (syntax) error
G	Reserved
H	User defined
I	Key identifier not known to recipient
K	Parity error in received key
M	MAC error (failure to authenticate)

O	Option not implemented
P	CTP error
T	CKT unknown
U	IDU not known to the CKT or CKD
X	EDC error (probable transmission error)

Error Service Messages shall be generated by computing or selecting field contents in accordance with the following process.

Process	Field	Action
1.	MCL	Insert ESM in the field. The field becomes: MCL/ESM
2.	RCV	Insert recipient's identity in the field. If RRRR is the identity, the field becomes: RCV/RRRR
3.	ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
4.	IDC	Error messages responding to a KSM or a RSM in a CKD or CKT environment shall return the IDC field contents. If the IDC field contents are CCCC, then the field becomes: IDC/CCCC
5.	IDU	All error messages responding to a ERS, RTR, or RSI in a CKD environment; or to a ERS, RFS or RTR in a CKT environment shall return the IDU field contents. If the IDU field contents are UUUU, then the field becomes: IDU/UUUU
6.	CTA	(Used in a CKD environment when responding to a RTR, or in a CKT environment when responding to a ERS or RFS) The count returned in this field shall contain the expected CTA (see Section 7.3). If the value is "a", the field becomes: CTA/a
7.	CTB	(Used in a CKD or CKT environment when responding to a KSM) The count returned in this field shall contain the expected CTB (see Section 7.3). If the value is "b", the field becomes: CTB/b

Process	Field	Action
8.	CTP	(Used only in a point-to-point environment when responding to a KSM) The count returned in this field shall contain the expected CTP (see Section 7.3). If the value is "p", the field becomes: CTP/p
9.	CTR	(Used when a count error has been detected) The count returned is the count included in the message to which this ESM responds. The following table identifies the situations when the CTR field shall be used in the message along with the count. The received count shall be copied from the previous message and inserted in the CTR field.

Environment	Previous Message	Count
Point-to-Point	KSM	CTP
CKD	KSM	CTB
CKD	RTR	CTA
CKT	KSM	CTB
CKT	RFS	CTA
CKT	ERS	CTA

10.	ERF	The contents of the ERF field are defined by the error conditions detected by the originator of this ESM. See the definition of ERF field contents in Section 8.6.1, Table II. Multiple error conditions are indicated by returning a concatenated string of error flags. E.g., ERF/KPM
11.	EDC	(Optional) The data key for EDC computation shall be: KDX = 0123456789ABCDEF The EDC is computed using: EDC = aKDX(MCL/ESMb ... bERF/KPMb) and if brackets are used to denote the field contents, the field becomes: EDC/[aKDX(MCL/ESMb ... bERF/KPMb)]

9.5 Generating A Key Service Message

A Key Service Message (KSM) may be generated spontaneously or in response to a RSI received from another party in a point-to-point environment (see Figures V and VI). In the CKD and CKT environments, however, the KSM is generated following receipt of a RTR from the CKD or CKT during a sequence of Cryptographic Service Message exchanges (see Figures VII-X).

The expected responses to a KSM are either a RSM or a ESM from the intended recipient of the KSM. If either message is not received within a predetermined period of time, an identical KSM may be sent for a given number of times. Key Service Messages shall be generated by computing or selecting field contents in accordance with the following process.

Process	Field	Action
1.	MCL	Insert KSM in the field to form: MCL/KSM
2.	RCV	Insert recipient's identity in the field. E.g., if RRRR is the identity, the field becomes: RCV/RRRR
3.	ORG	Insert originator's identity in the field. E.g., if OOOO is the identity, the field becomes: ORG/OOOO
4.	IDC	(CKD or CKT environment only) Insert the identity of the CKD or CKT used in the key distribution process of which this KSM is a step. E.g., if CCCC is the identity, the field becomes: IDC/CCCC
5.	NOS	(point-to-point environment only) (Optional) If notarization of the (*)KK (or KDs if no (*)KK is sent in the message) is desired, include the NOS field in the KSM. The field becomes: NOS/
6.	(*)KK	(only used in a point-to-point environment) (optional)
	Optional Subfields	If it is desired to use the odd parity feature, to name the (*)KK being sent in the Cryptographic Service Message or to specify the key encrypting key to be used to decrypt the (*)KK (or any combination thereof), form and insert the applicable subfields using the rules of section 8.5, above.

Process	Field	Action
		If a (*)KK is sent, an associated count shall be established for key offsetting the (*)KK when other keys are received which are encrypted using this (*)KK. The count is initially set to one, and the value shall be used to key offset and encrypt the KD which is sent in this message. The (*)KK used to encrypt the (*)KK sent in the Cryptographic Service Message shall be a key currently shared with the message recipient.
		Case 1: KK to be sent encrypted by a KK, no notarization
		If a new KK is to be sent, then for a single length KK, the encrypted KK is computed using the following equation:
		Let KKZ be the key to be encrypted and KKY be the encrypting key.
		Use the procedure of Section 7.4 and the value of CTP to compute the KKoY.
		Encrypt KKZ
		encrypted KKZ = eKKoY(KKZ)
		If brackets are used to denote the field contents, then the field becomes:
		KK/[eKKoY(KKZ) (optional subfields)]
		Case 2: (*)KK to be sent encrypted by a *KK, no notarization
		Use the procedure of Section 7.4 and the value of CTP to compute the *KKoY.
		Encrypt (*)KKZ
		Encrypted KKZ = ede*KKoY(KKZ) or
		Encrypted *KKZ = ede*KKoY(*KKZ)
		If brackets are used to denote field contents, the field becomes:
		KK/[ede*KKoY(KKZ) (optional subfields)] or
		*KK/[ede*KKoY(*KKZ) (optional subfields)]
		Case 3: (*)KK with notarization
		The (*)KK field contents are computed using the following equations:
	(a)	Compute (*)KN using the contents of the ORG, RCV, and CTP fields and the process defined in Section 7.5, above.

Process	Field	Action
	(b)	Encrypt the (*)KKZ to form the contents of the (*)KK field using the equations: notarized KKZ = eKN(KKZ) or notarized KKZ = ede*KN(KKZ) or notarized *KKZ = ede*KN(*KKZ) respectively.
	(c)	If brackets are used to denote field contents, the field becomes: KK/[eKN(KKZ) (optional subfields)] or KK/[ede*KN(KKZ) (optional subfields)] or *KK/[ede*KN(*KKZ) (optional subfields)] respectively.
7.	(*)KKU	(CKT environment only) (Optional) Let (*)KKUC be the contents of the (*)KKU field received in the RTR message from which the KSM is generated. Then the contents of the (*)KKU field in the KSM are set equal to (*)KKUC and the field becomes: (*)KKU/[(*)KKUC (optional subfields)] If a (*)KKU is sent, an associated count shall be established for key offsetting the (*)KK when other keys are sent using this (*)KKU. The count shall be initially set to one, and the value shall be used to key offset encrypt the KD which is sent in this message.
8.	KD	(Not used in a CKD environment) At least one and at most two KDs shall be sent in a KSM as KD field(s). In a CKT environment, KD field(s) shall be present if and only if a (*)KKU field is contained in this Cryptographic Service Message. If a new (*)KK is sent in the KSM, then the KDs shall be encrypted using that key.
	Optional Subfields	If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message or to specify the key encrypting key to be used to decrypt the KDs (or any combination thereof), form and insert the applicable subfields using the rules of Section 8.5, above.

Process	Field	Action
		KDS shall be encrypted or given notarization protection using the processes that follow.
		Case 1: KD encrypted by a (*)KK; no notarization
		KDs are encrypted by a (*)KK and are not notarized (i.e., there is no NOS field in the KSM or a (*)KK or (*)KKU is sent). If a (*)KK or (*)KKU is sent in this Cryptographic Service Message, this (*)KK shall be used to encrypt the KDs. Otherwise, a currently shared (*)KK is used.
		Let KDI be the key to be sent in this field. Let (*)KKY be the key encrypting key to be used. Then the KD(s) is (are) computed using the equations:
	(a)	Use the procedure of Section 7.4 to compute the (*)KKoY:
		using CTP in a point-to-point environment if no (*)KK is sent in the message and a value of one (1) if a new (*)KK is sent. In a CKT environment the count shall be set equal to a value of one (1).
	(b)	Encrypt the KDI using the equation:
		encrypted KDI = eKKoY(KDI) or
		encrypted KDI = ede*KKoY(KDI)
	(c)	If brackets are used to denote field contents, the field becomes:
		KD/[eKKoY(KDI) (optional subfields)] or
		KD/[ede*KKoY(KDI) (optional subfields)]
		Case 2: KD notarized under a (*)KK (only in a point-to-point environment)
		There is no (*)KK field in the message and the KDs are to be given notarization protection. In this case, the NOS field shall be included in the KSM and the KD fields content shall be computed using the following equations:
	(a)	Compute (*)KN using the contents of the ORG, RCV and CTP fields and the process defined in Section 7.5.
	(b)	Encrypt the KDI to form the contents of the KD field using the equations:
		notarized KDI = eKN(KDI) or
		notarized KDI = ede*KN(KDI)
		for a KN or *KN, respectively.

Process	Field	Action
	(c)	If brackets are used to denote field contents, the field becomes: KD/[eKN(KDI) (optional subfields)] or KD/[ede*KN(KDI) (optional subfields)] for a KN or *KN, respectively.
9.	KDU	(CKD or CKT environment only) At least one and at most two KDS shall be sent in a KSM as KDU field(s). This field shall be present in a KSM if and only if there is no (*)KKU field in the Cryptographic Service Message. Let KDUC be the contents of the KDU field received in the RTR from which this KSM is generated. Then the contents of the KDU field in the KSM are set equal to KDUC and the field becomes: KDU/[KDUC (optional subfields)]
10.	IV	(Optional) Case 1: Encrypted IV (point-to-point and CKT environments only) If an IV is sent in encrypted form, the IV shall be encrypted using the KD sent in the Cryptographic Service Message (the second KD if two are sent) using the equation: encrypted IV = eKD(IV) and the field is: IV/[E | | eKD(IV)] Case 2: Plaintext IV (point-to-point and CKT environments only) If an IV is sent in plaintext form, then the field is: IV/[P | | IV] Case 3: Encrypted or Plaintext IV (CKD environment only) Let IVC be the contents of the IV field received in the RTR from which this KSM is generated, if present, otherwise the IV (IVC) may be determined by the originating party. Then the contents of the IV field in the KSM are set equal to IVC and the field becomes: IV/IVC

Process	Field	Action
11.	EDK	(Optional) Let YYMMDDHHMMSS be the contents of the EDK field received in the RTR from which this KSM is generated, if present, otherwise the EDK field may be determined by the originating party. Then the contents of the EDK field shall be: YYMMDDHHMMSS and the field becomes: EDK/[YYMMDDHHMMSS]
12.	CTB	(CKD or CKT environment only) CTB is the value of the counter shared by the CKD or CKT and the ultimate recipient. This CTB field is formed by setting it equal to the CTB field value received in the RTR from the CKD or CKT which initiated the generation of this KSM. If the value received in the CTB field of the RTR is "b", then the CTB field is: CTB/b
13	CTP	(point-to-point environment only) If the value of the CTP before KSM preparation is p, then the KSM shall contain the CTP field: CTP/p
14.	MAC	The MAC is always computed using the KDs sent in the message. If only one KD is sent, KDJ, then that key shall be used. If two KDs are sent (KDH and KDI) then the key, KDJ (used to authenticate the Cryptographic Service Message) is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ(MCL/KSMb...bCTX/xb) and if brackets are used to denote the field contents, the field becomes: MAC/[aKDJ(MCL/KSMb...bCTX/xb)] where CTX is CTP and x is p in a point-to-point environment; and CTX is CTB and x is b in a CKD or CKT environment.

Process	Field	Action
		Input to the authentication algorithm starts with the first character following the open parenthesis "(" of the Cryptographic Service Message, and continues through the blank, "b", following the CTX field, inclusive.

9.6 Generating A Request For Service Message

A Request For Service Message (RFS) is only sent to a CKT (see Figures IX-X). The RFS may be initiated by the originating party or may be generated following receipt of a RSI.

All keys and IVs in a RFS shall be generated by the originator and sent to the CKT. Request for Service Messages shall be generated by computing or selecting field contents in accordance with the following process.

Process	Field	Action
1.	MCL	Insert RFS in the field to form: MCL/RFS
2.	RCV	Insert recipient's identity (i.e., the identity of the CKT) in the field. If RRRR is the identity, the field becomes: RCV/RRRR
3.	ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
4.	IDU	Insert the identity of the ultimate recipient in the field. If UUUU is the identity, the field becomes: IDU/UUUU
5.	(*)KK	(Optional)
	Optional Subfields	
		If it is desired to use the odd parity feature, to name the (*)KK being sent in the Cryptographic Service Message or to specify the key encrypting key to be used to decrypt the (*)KK (or any combination thereof), form and insert the applicable subfields using the rules of Section 8.5, above.
		If a (*)KK is to be sent, the (*)KK shall be generated or acquired and an associated count shall be established for key offsetting the (*)KK when other keys are encrypted using this (*)KK. This count is initially set to zero, and the value zero shall be used to key offset encrypt the KD to be sent in this Cryptographic Service Message.

Process	Field	Action
		Let (*)KKZ be the key to be encrypted (i.e., to be sent to the IDU party) and *KKY be the encrypting key shared with the CKT.
		If a new (*)KK is to be sent, then the encrypted (*)KK is computed using the following equation:
		Use the procedure of Section 7.4 to compute the *KKoY
		Then:
		encrypted KKZ = ede*KKoY(KKZ) or
		encrypted *KKZ = ede*KKoY(*KKZ)
		If brackets are used to denote the field contents, then the field becomes:
		KK/[ede*KKoY(KKZ) (optional subfields)] or
		*KK/[ede*KKoY(*KKZ) (optional subfields)]
6.	KD	KDs sent in a RFS to a CKT are encrypted data keys that are used to authenticate that RFS. If no (*)KK field is present, KDs are sent to the ultimate recipient as KDUs. Note that KDs are never given notarization protection in a RFS.
	Optional Subfields	
		If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message or to specify the key encrypting key to be used to decrypt the KDs (or any combination thereof), form and insert the applicable subfields using the rules of Section 8.5.
		Case 1: The RFS contains a (*)KK field
		One and only one KD shall be sent in the RFS in the KD field. That KD shall be used to authenticate the RFS and to generate the MAC for inclusion in the RTR that responds to this RFS.
		Let KDI be the key to be sent in this field. Let (*)KKY be the key encrypting key sent in the message. The KD is computed using the equations:
		Use the procedure of Section 7.4 and an offset of zero to compute (*)KKoY.
	(a)	Encrypt the KDI using the equation:
		encrypted KDI = eKKoY(KDI) or
		encrypted KDI = ede*KKoY(KDI)

Process	Field	Action
	(b)	If brackets are used to denote field contents, the field becomes: KD/[eKKoY(KDI) .P] or KD/[ede*KKoY(KDI) .P] where P is an optional subfield. Case 2: There is no (*)KK field in the message. At least one and at most two KDs shall be sent in a RFS as KD field(s). A KD shall be encrypted by a *KK, and the following equation is used:
	(a)	Use the procedure of Section 7.4 and the value of CTA to compute the *KKoY.
	(b)	Encrypt the KDI using the equation: encrypted KDI = ede*KKoY(KDI)
	(c)	If brackets are used to denote field contents, the field becomes: KD/[ede*KKoY(KDI) (optional subfields)]
7.	CTA	If the value of the CTA before RFS preparation is "a", then the RFS shall contain the CTA field: CTA/a
8.	MAC	The MAC is always computed using the KDs sent in the Cryptographic Service Message. If only one KD is sent, KDJ, then that key shall be used. If two KDs are sent, KDH and KDI, then the key KDJ (used to authenticate the Cryptographic Service Message) is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ(MCL/RFSb...bCTA/ab) and if brackets are used to denote field contents, the field becomes: MAC/[aKDJ(MCL/RFSb...bCTA/ab)] Input to the authentication algorithm starts with the first character following the open parenthesis "(" of the Cryptographic Service Message, and continues through the blank, "b", following the CTA field, inclusive.

9.7 Generating A Request Service Initiation Message

A Request Service Initiation Message (RSI) is generated by the originating party in all environments in order to request that keys be sent in a subsequent KSM to establish a keying relationship (see Figures V-X).

In the CKD environment, the RSI is sent to the CKD to request keys which shall be sent to another party (the ultimate recipient) in a later KSM (see Figures VII-VIII).

Request Service Initiation Messages shall be generated by computing or selecting field contents in accordance with the following process.

Process	Field	Action
1.	MCL	Insert RSI in the field to form: MCL/RSI
2.	RCV	Insert recipient's identity in the field. If RRRR is the identity, the field becomes: RCV/RRRR
3.	ORG	Insert originator's identity in the field. if OOOO is the identity, the field becomes: ORG/OOOO
4.	IDU	(Required only in a CKD environment when the RSI is sent to the center) Insert the identity of the ultimate recipient in the field. If UUUU is the identity, the field becomes: IDU/UUUU
5.	IDC	(Required in a RSI from Party B to Party A in a center environment (only)) Insert the identity of the CKD or CKT in the field. If CCCC is the identity, then the field becomes: IDC/CCCC
6.	SVR	Insert the subfields designating the type of service requested. Note that a single data key is implicitly requested by the presence of a SVR field. E.g., SVR/ to request one data key SVR/KD to request two data keys SVR/KK to request a single data key and a key encrypting key

Process	Field	Action
		SVR/KK.KD.IV to request a key encrypting key, two data keys and an encrypted IV
		SVR/*KK.KD.IV to request a key encrypting key pair, two data keys and an encrypted IV
		A (*)KK shall not be requested in a CKD environment.
7.	EDC	(Optional)
		The data key for EDC computation shall be:
		KDX = 0123456789ABCDEF
		The EDC is computed using:
		EDC = aKDX(MCL/RSIb...bSVR/*KK.KD.IVb)
		and the field becomes:
		EDC/[aKDX(MCL/RSIb...bSVR/*KK.KD.IVb)]

9.8 Generating A Response Service Message

A Response Service Message (RSM) is generated following receipt of a acceptable DSM or KSM. Response Service Messages shall be generated by computing or selecting field contents in accordance with the following process.

Process	Field	Action
1.	MCL	Insert RSM in the field to form:
		MCL/RSM
2.	RCV	Insert recipient's identity in the field. If RRRR is the identity, the field becomes:
		RCV/RRRR
3.	ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes:
		ORG/OOOO
4.	IDC	(CKD or CKT environment only, when responding to a KSM)
		Insert the identity of the CKD or CKT used in the key distribution process of which this RSM is a step. E.g., if CCCC is the identity, the field becomes:
		IDC/CCCC
5.	IDD	(only used in response to a DSM)

Process	Field	Action
		Copy the IDD field(s) from the DSM to which this RSM responds.
6.	IDU	(In response to a unsolicited RTR; CKD environment only) Insert the identity of the ultimate recipient in the field. If the identity of the ultimate recipient is UUUU, then the field becomes: IDU/UUUU
7.	MAC	The MAC is always computed using the KDs sent or specified in the DSM, KSM or RTR to which the RSM responds. If only one KD is sent or specified, KDJ, then that key shall be used. When responding to a DSM, the key, KDJ, shall be the key identified in the IDA field or the only data key shared between the originating and recipient parties if that data key is unnamed. If two KDs are sent (KDH and KDI) then the key, KDJ (used to authenticate the Cryptographic Service Message) is derived from the equation: KDJ = (KDH + KDI) The MAC is then:
	(a)	aKDJ(MCL/RSMb...bIDD/KKKKb) responding to a DSM
	(b)	aKDJ(MCL/RSMb...bIDC/CCCCb) responding to a KSM in a center environment
	(c)	aKDJ(MCL/RSMb...bORG/OOOOb) responding to a KSM in a point-to-point environment
	(d)	aKDJ(MCL/RSMb...bIDU/UUUUb) responding to a unsolicited RTR in a CKD environment
		and if brackets are used to denote field contents, the field becomes:
	(a)	MAC/[aKDJ(MCL/RSMb...bIDD/KKKKb)] responding to a DSM
	(b)	MAC/[aKDJ(MCL/RSMb...bIDC/CCCCb)) responding to a KSM in a center environment
	(c)	MAC/[aKDJ(MCL/RSMb...bORG/OOOOb)]

Process	Field	Action
		responding to a KSM in a point-to-point environment
	(d)	MAC/[aKDJ(MCL/RSMb...bIDU/UUUUb)]
		responding to a unsolicited RTR in a CKD environment
		Input to the authentication algorithm starts with the first character following the open parenthesis "(" of the Cryptographic Service Message, and continues through the blank "b" preceding the MAC field tag.
		When the RSM has been generated in response to a DSM, the key(s) identified in the IDD field(s) of the DSM (or used to compute the MAC if the IDD field is null and no IDA field is present) shall be discontinued.

9.9 Generating A Response To Request Message

A Response To Request Message (RTR) is generated in response to a RFS (CKT environment), a RSI (CKD environment), or a ERS (either environment) (see Figures VII-X). A unsolicited RTR may be generated in a CKD environment.

Response To Request Service Messages shall be generated by computing or selecting field contents in accordance with the following process.

Process	Field	Action
1.	MCL	Insert RTR in the field to form
		MCL/RTR
2.	RCV	Insert recipient's identity in the field. If RRRR is the identity, the field becomes
		RCV/RRRR
3.	ORG	Insert originator's identity in the field. If OOOO is the identity of the CKD or CKT, the field becomes
		ORG/OOOO
4.	IDU	Insert the identity of the ultimate recipient in the field. If the identity of the ultimate recipient is UUUU, then the field becomes:
		IDU/UUUU
5.	(*)KKU	(CKT environment only) (optional)
		This field is present if and only if there was a (*)KK field present in the RFS or ERS to which this RTR responds.

Process	Field	Action
	Optional Subfields	
		If it is desired to use the odd parity feature, to name the (*)KK being sent in the Cryptographic Service Message or to specify the key encrypting key to be used to decrypt the (*)KK (or any combination thereof), form and insert the applicable subfields using the rules of Section 8.5.
		Let *KKY be the *KK shared between Party B and the CKT. Let (*)KKZ be the (*)KK that is to be sent to Party B, the ultimate recipient. The (*)KKZ is the key encrypting key received from Party A in the RFS to which this RTR responds.
		The (*)KKZ shall be notarized before transmission.
		The (*)KKU field contents shall be computed using the following equations:
(a)		Compute *KN using *KKY and the contents of the RCV, IDU and CTB fields and the process defined in Section 7.5.
(b)		Encrypt the (*)KKZ to form the contents of the (*)KKU field using the equations:
		KKU notarized KKZ = ede*KN(KKZ) or
		*KKU notarized *KKZ = ede*KN(*KKZ)
(c)		If brackets are used to denote field contents, the field becomes, e.g.,
		KKU/[ede*KN(KKZ) (optional subfields)] or
		*KKU/[ede*KN(*KKZ) (optional subfields)]
6.	KD	(CKD environment only)
		At least one and at most two KDs shall be generated and sent in a RTR Message as KD field(s).
	Optional Subfields	
		If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message or to specify the key encrypting key to be used to decrypt the KDs (or any combination thereof), form and insert the applicable subfields using the rules of Section 8.5.
		KDs shall be encrypted using the process that follows.
		Let KDI be the key to be sent in this field. Let *KKY be the key encrypting key shared by Party A and the CKD to be used.

Process	Field	Action
		The following equation is used:
	(a)	Compute the *KN using *KKY and the contents of the RCV, IDU and CTA fields in this RTR, and the process defined by Section 7.5.
	(b)	Encrypt the KDI using the equation: encrypted KDI = ede*KN(KDI)
	(c)	If brackets are used to denote field contents, the field becomes: KD/[ede*KN(KDI) (optional subfields)]
7.	KDU	If and only if no (*)KK is being sent as a (*)KKU, then at least one and at most two notarized KDs for the ultimate recipient shall be sent as KDU field(s).
	Optional Subfields	
		If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message or to specify the key encrypting key to be used to decrypt the KDs (or any combination thereof), form and insert the applicable subfields using the rules of Section 8.5. Let KDJ be a data key that is to be sent to the ultimate recipient. The KDU field(s) contents shall be computed using the following equations:
	(a)	Compute *KN using the key encrypting key, *KKY, shared by Party B and the CKD, the contents of the RCV, IDU and CTB fields, and the process defined in Section 7.5.
	(b)	Encrypt the KD to form the contents of the KDU field using the equation: KDU = notarized KD = ede*KN(KD)
	(c)	If brackets are used to denote field contents, the field becomes: KDU/[ede*KN(KD) (optional subfields)]
8.	IV	(CKD environment only) (Optional) Case 1: Encrypted IV If an IV is to be sent in encrypted form, the IV shall be encrypted as follows: Let KDH be the KD sent in this Cryptographic Service Message. If two KDs are sent, the second KD (used for encryption) shall be used.

Process	Field	Action
		encrypted IV = eKDH(IV)
		and the field is:
		IV/[E | | eKDH(IV)]
		Case 2: Plaintext IV
		If an IV is sent in plaintext form, then the field is:
		IV/[P | | IV]
9.	EDK	(CKD environment only) (Optional)
		If an EDK is sent, the field shall be
		EDK/[YYMMDDHHMMSS]
10.	CTB	If the value of the CTB field before RTR preparation is "b", then the RTR shall contain the CTB field:
		CTB/b
11.	CTA	(CKD environment only)
		CTA is the value of the counter associated with the *KK shared by the originator of the RSI or ERS to which this RTR responds and the CKD.
		If the value of CTA before RTR preparation is "a", then the RTR shall contain the CTA field:
		CTA/a
12.	MAC	In a CKT environment, when a (*)KKU is returned in the RTR, the KD used to authenticate the RTR shall be the KD received for authenticating the RFS. In a CKD environment, the KDs used in the MAC computation are the KDs being sent in the message.
		If only one KD is sent, KDJ, then that key shall be used. If two KDs are sent (KDH and KDI) then the key, KDJ (used to authenticate the Cryptographic Service Message), is derived from the equation:
		KDJ = (KDH + KDI)
		The MAC is then:
		aKDJ(MCL/RTRb...bCTX/xb)
		and if brackets are used to denote field contents, the field becomes:

Process	Field	Action
		MAC/[aKDJ(MCL/RTRb...bCTX/xb)]
		Where CTX is CTA and "x" is "a" in a CKD environment; and CTX is CTB and "x" is "b" in a CKT environment.
		Input to the authentication algorithm starts with the first character following the open parenthesis "(" of the Cryptographic Service Message, and continues through the blank "b" following the CTX field, inclusive.

10. Processing Cryptographic Service Messages

10.1 Cryptographic Service Message Class Determination

The class of the incoming Cryptographic Service Message is determined by examining the first field (the three letters following "CSM(MCL/"). The following table references the section of this standard which shall be used in processing each class of Cryptographic Service Message.

If the MCL Field Content Is:	The Cryptographic Service Message Shall Be Processed in Accordance With:	Page Number:
DSM	Section 10.2	81
ERS	Section 10.3	82
ESM	Section 10.4	87
KSM	Section 10.5	90
RFS	Section 10.6	98
RSI	Section 10.7	102
RSM	Section 10.8	104
RTR	Section 10.9	107

If the MCL field contains a value other than those listed above, an error condition exists. A ESM shall be sent with an "F" in the ERF field. I.e.,

ERF/F

If the identity of a party sending a Cryptographic Service Message is not known to the recipient, a ESM may be sent or the problem may be resolved by other means. If a ESM is sent, it shall have a "C" in the ERF field. I.e.,

ERF/C

10.2 Processing A Disconnect Service Message

A Disconnect Service Message (DSM) notifies the recipient of the DSM that one or more keys are to be terminated. Responses to the DSM are either a RSM if the DSM is received with no errors, or a ESM if errors are detected in the DSM.

Disconnect Service Messages shall be processed by computing or selecting field contents in accordance with the following steps.

Process	Field	Action
1.	MCL	Examine the contents of this field to determine the Cryptographic Service Message class.
2.	RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the DSM for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
3.	ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
4.	IDD	If an IDD field is null and the DSM processes correctly, then the keying relationship shall be terminated. If not null, the IDD field contains the identity of a (*)KK or KD to be discontinued. The IDD field(s) shall be inserted into the RSM generated in response to this Cryptographic Service Message. If the IDD is not known to the recipient, this shall cause processing of the DSM to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field. I.e., ERF/I
5.	IDA	The IDA is the identity of the KD used to compute the MAC. This same KD shall be used to authenticate the RSM generated in response to this Cryptographic Service Message. If the IDA field is not present, the data key to be used in computing the MAC shall be the only data key shared between the originating and recipient parties. The key named in the IDA field (if present) or the only data key shared by the two parties shall be discontinued after generation of the RSM that responds to this DSM even if it is erroneously not named in the IDD field.

Process	Field	Action
		If the key named by the IDA field is not known to the recipient, this shall cause processing of the DSM to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field. I.e., ERF/I
6.	MAC	Compute a MAC from the message. The KD that shall be used in the MAC computation is the KD identified in the IDA field. The MAC is then: aKD(MCL/DSMb...bIDA/IDK1b) If the computed MAC does not equal the received MAC, the message fails to authenticate. An ESM may be generated and returned to the originator. The ERF field shall include an "M". I.e., ERF/M Alternatively, the error may be resolved by manual means (e.g., telephone).

10.3 Processing An Error Recovery Service Message

An Error Recovery Service Message (ERS) is received by a CKD or CKT which:

(1) announces that a problem exists in the key or count shared by the CKD or CKT and the party identified by the IDU field (ultimate recipient), and

(2) requests that further keys be processed (after appropriate corrections are made) and sent to the originator of the ERS in a RTR. The keys shall then be forwarded to the ultimate recipient in a KSM (see Figures VIII, X).

Error Recovery Service Messages shall be processed by computing or selecting field contents in accordance with the following steps.

Process	Field	Action
1.	MCL	Examine the contents of this field to determine the Cryptographic Service Message class.
2.	RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient.

Process	Field	Action
		If RRRR is not the identity of the party receiving the ERS for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
3.	ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
4.	IDU	This field contains the identity of the ultimate recipient. If the field is IDU/UUUU, the ultimate recipient is UUUU. The field contents shall be used in selecting the *KK to be used in generating the response to this Cryptographic Service Message. If this identity is not known, a ESM shall be sent with a "U" in the ERF field. Further Cryptographic Service Message processing may be performed in order to check keys, counts and message authentication prior to actual transmission of the ESM.
5.	(*)KK	(CKT environment only) (When present) Using the rules of Section 8.5, parse the field to obtain the (*)KK; and the P, IDK1 and IDK2 subfields if present. If the IDK2 subfield is not present, the *KK used to decrypt the received (*)KK is the only *KK shared by the message originator and the CKT.
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If on decryption, the key does not conform to the specification for odd parity, a ESM shall be generated in response to the ERS, with a "K" in the ERF field. All further Cryptographic Service Message processing shall cease. The "P" shall be inserted in the "P" subfield of the (*)KKU field in the RTR generated in response to this Cryptographic Service Message using the rules of Section 8.5.
	IDK1	If present, the IDK1 name shall be inserted in the IDK1 subfield of the (*)KKU field in the RTR generated in response to this Cryptographic Service Message, using the rules of Section 8.5. Otherwise, no IDK1 subfield shall be sent in the (*)KKU field in the RTR.
	IDK2	If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the ERS to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field, i.e.,

Process	Field	Action
		ERF/I
		Decryption of (*)KK
		The decrypted (*)KK is computed using the following equations:
		Let (*)KKZ be the key to be decrypted and *KKY be the decrypting key.
		Use the procedure of Section 7.4 and the contents of the CTA field to compute the *KKoY.
		Decrypt (*)KKZ
		decrypted KKZ = ded*KKoY(encrypted KKZ) or
		decrypted *KKZ = ded*KKoY(encrypted *KKZ)
6.	KD	(CKT environment only)
		At least one and at most two KDs shall be received in a ERS as KD field(s).
		If a new (*)KK is sent in the ERS, then only one KD shall be received in the ERS and that KD shall be decrypted using that (*)KK.
		The KD is to be decrypted by a *KK currently shared with the message originator. The KD field may have subfields as defined in Section 8.5. Select the *KK as defined below.
		Parse the KD field(s) to obtain the KD; and the P, IDK1, and IDK2 subfields if present.
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If, on decryption, the key does not conform to the specification for odd parity, a ESM shall be generated in response to the ERS, with a "K" in the ERF field. I.e.,
		ERF/K
		Further processing of the ERS shall cease.
		If the key parity is correct, the "P" shall be inserted in the "P" subfield of the RTR generated in response to this Cryptographic Service Message using the rules of Section 8.5.

Process	Field	Action
	IDK1	If present, this subfield names the KD. The IDK1 name shall be inserted in the IDK1 subfield of the KDU field in the RTR generated in response to this Cryptographic Service Message, using the rules of Section 8.5.
	IDK2	The *KK used to decrypt the KD is identified by the IDK2 subfield, if present. See Section 8.5.
		If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the ERS to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field, i.e.,
		ERF/I
		Decryption of the KDs:
		Case 1: The ERS contains a (*)KK field
		One and only one KD shall be received in the ERS in the KD field. That KD shall be used to authenticate the ERS and to generate the MAC for inclusion in the RTR that responds to this ERS.
		Let KDI be the key received in this field. Let (*)KKY be the key encrypting key sent in the Cryptographic Service Message. The KD is decrypted using the equations:
	(a)	The KD is decrypted by an (*)KK; an offset of zero is used.
	(b)	Decrypt the KDI using the equation:
		decrypted KDI = dKKoY(KDI) or
		decrypted KDI = ded*KKoY(KDI)
		Case 2: There is no (*)KK field in the Cryptographic Service Message.
		At least one and at most two KDs shall be received in a ERS as KD field(s).
		The KDs are decrypted by a *KK, and the following equations are used:
	(a)	Use the procedure of Section 7.4 to compute the *KKoY
	(b)	Decrypt the KDI using the equation:
		decrypted KDI = ded*KKoY(KDI)

Process	Field	Action
7.	ERF	Parse the field to obtain the codes for the types of errors reported. These error types shall be utilized in generating the RTR that responds to this ERS or in the manual recovery process, if necessary.
8.	SVR	(CKD environment only) Parse the field to obtain the designators for the types of service requested. These service types shall be utilized in generating the RTR that responds to this ERS. Note that a single data key is implicitly requested by the presence of a SVR field. See Table II for a definition of the types of service requested.
9.	CTB	The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of CTB field. If the field is CTB/b, the CTB value is "b".
10.	CTR	The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. The value in the CTR field may be used in determining the Cryptographic Service Message to which this ERS responds. If the field is CTR/r, the CTR value is "r".
11.	CTA	(CKT environment only) The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of the CTA field. If the field is CTA/a, the CTA value is "a". If a CTA error is detected, a ESM shall be generated to notify the originating party of the CTA error condition. The ESM shall have an "A" in the ERF field. I.e., ERF/A Further processing of the ERS may continue prior to transmission of the ESM.
12.	MAC	(CKT environment only) Compute a MAC from the message. The MAC shall always be computed using the KDs received in the message. If only one KD is received, KDJ, then that key shall be used. If two KDs are received (KDH and KDI), then the key, KDJ (used to authenticate the Cryptographic Service Message), is derived from the equation:

Process	Field	Action
		KDJ = (KDH + KDI) The MAC is then: aKDJ(MCL/ERSb...bCTA/ab) If the computed MAC does not equal the received MAC, the message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include an "M". I.e., ERF/M
13.	EDC	(CKD environment only) (when present) If this option is not implemented, a ESM may be generated and returned to the originator with an "O" in the ERF field. I.e., ERF/O Alternatively, the field may be disregarded and message processing may proceed. Compute an EDC from the message using the data key for EDC computation (KDX = 0123456789ABCDEF): EDC = aKDX(MCL/ERSb...bCTB/bb) If the computed EDC does not equal the received EDC, there is an error (possibly a transmission error). A ESM shall be generated and returned to the originator. The ERF field shall include an "X" I.e., ERF/X

10.4 Processing An Error Service Message

An Error Service Message (ESM) is received in response to a DSM, ERS, KSM, RSI, RSM, RFS or RTR due to the detection of one or more of the following error conditions:

Error Code	Definition
A	CTA error
B	CTB error
C	Cannot process
D	CKD unknown
E	Facility inoperative
F	Format (syntax) error
G	Reserved

H	User defined
I	Key identifier not known to recipient
K	Parity error in received key
M	MAC error (failure to authenticate)
O	Option not implemented
P	CTP error
T	CKT unknown
U	IDU not known to the CKT or CKD
X	EDC error (probable transmission error)

Error Service Messages shall be processed as follows:

Process	Field	Action
1.	MCL	Examine the contents of this field to determine the Cryptographic Service Message class.
2.	RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the ESM for processing, then the message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
3.	ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
4.	IDC	(Used in a center environment when responding to a KSM or RSM) This field contains the identity of the CKD or CKT. If the field is IDC/CCCC, the Center is CCCC.
5.	IDU	(Used in a CKD environment when responding to a ERS, RSI (to a CKD) and RTR, and in a CKT environment when responding to a ERS, RFS and RTR) This field contains the identity of the ultimate recipient. If the field is IDU/UUUU, the ultimate recipient is UUUU. If the identity of the ultimate recipient is unknown, a ESM with a "U" in the ERF field, i.e., ERF/U may be generated or manual recovery may be used. Further processing of the message may be performed prior to transmission of the error message.

Process	Field	Action
6.	CTA	(Used in a CKD environment when responding to a RTR and in a CKT environment when responding to a ERS or RFS) The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of CTA field. If the field is CTA/a, the CTA value is "a".
7.	CTB	(CKD or CKT environment; only when responding to a KSM) The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of CTB field. If the field is CTB/b, the CTB value is "b". This value (i.e., "b") shall be sent as the contents of the CTB field in the ERS that is generated in response to this ESM.
8.	CTP	(used only in a point-to-point environment when responding to a KSM) The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of CTP field. If the field is CTP/p, the CTP value is "p".
9.	CTR	The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of the CTR field. If the field is CTR/r, the CTR value is "r". The value of CTR may be used in determining the Cryptographic Service Message to which this ESM responds.
10.	ERF	Parse the field to obtain the designators for the type(s) of errors reported. These error type(s) shall be utilized in generating the Cryptographic Service Message that responds to this ESM or in the manual recovery process, if necessary. See the definition of ERF field contents in Table II. Multiple error conditions are indicated by a string of concatenated error flags. E.g., ERF/KPM
11.	EDC	(When present) If this option is not implemented, a ESM may be generated and returned to the originator with an "O" in the ERF field. I.e.,

Process	Field	Action
		ERF/O
		Alternatively, the field may be disregarded and message processing may proceed.
		Compute an EDC from the message using the data key for EDC computation (KDX = 0123456789ABCDEF):
		EDC = aKDX(MCL/ESMb...bERF/KPMb)
		If the computed EDC does not equal the received EDC, there is an error (possibly a transmission error). A ESM shall be generated and returned to the originator. The ERF field shall include an "X". Alternatively, the matter may be resolved by other means (e.g., telephone). I.e.,
		ERF/X

10.5 Processing A Key Service Message

A Key Service Message (KSM) is received from a party in order to establish a keying relationship and begin communications. Alternatively, a KSM may be sent to initiate a key change in an existing relationship (point-to-point environment only). Responses to the KSM are either a RSM if the KSM is received with no errors, or a ESM if errors are detected in the KSM.

Key Service Messages shall be processed as follows:

Note: If the "P" subfield is present and if there is a parity error in a decrypted key, then no further processing of the Cryptographic Service Message shall be done. An ESM shall be generated and sent to the originator of the KSM. The contents of the ERF field shall include, but not be limited to "K". E.g., the ERF field might be ERF/K.

Process	Field	Action
1.	MCL	Examine the contents of this field to determine the Cryptographic Service Message class.
2.	RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient.
		If RRRR is not the identity of the party receiving the KSM for processing, then the message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
3.	ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.

Process	Field	Action
4.	IDC	(CKD or CKT environment only) Process the value of the IDC field. If the field is IDC/CCCC, the value is CCCC. This is the identity of the CKD or CKT, and is used to select the keys and other data used to process the message. If this identity is not known, a ESM shall be sent with a "D" (CKD environment) or a "T" (CKT environment) in the ERF field. Further processing of the message shall cease.
5.	NOS	(only used in a point-to-point environment) (When present) If this field is present, the (*)KK (or KDs if no (*)KK is sent in the message) has (have) been notarized.
6.	(*)KK	(only used in a point-to-point environment) (When present) Using the rules of Section 8.5, parse the field to obtain the (*)KK; and the P, IDK1 and IDK2 subfields if present. If an IDK2 subfield is not present, the (*)KK used to decrypt the received (*)KK is the only one shared by the message originator and recipient.
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If on decryption, the key does not conform to the specification for odd parity, a ESM shall be generated in response to the KSM, with a "K" in the ERF field. All further processing of the message shall cease.
	IDK1	If a (*)KK is received in the message, and no IDK1 field is present, then, immediately on decryption of the new (*)KK, that key shall be placed into use for all subsequent processing.
	IDK2	The IDK2 subfield (if present) defines the (*)KK, (*)KKY, to be used in decrypting the (*)KK contained in the KSM. Otherwise, the (*)KK to be used for decryption of the received (*)KK is implicitly defined. See Section 8.5. If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field, i.e.,

Process	Field	Action
		ERF/I
		If a (*)KK is received, an associated count shall be established for key offsetting the (*)KK when other keys are received which are encrypted using this (*)KK. The count (CTP) is initially set to one, and the value one shall be used to key offset decrypt the KD which is received in this message.
		Decryption of (*)KKs
		Case 1: (*)KK, without notarization
		If a new (*)KK is received, then the decrypted (*)KK is computed using the following equations:
		Let (*)KKZ be the key to be decrypted and (*)KKY be the decrypting key.
		Use the procedure of Section 7.4 and the value of CTP to compute the (*)KKoY:
		Decrypt (*)KKZ
		decrypted KKZ = dKKoY(encrypted KKZ) or
		decrypted KKZ = ded*KKoY(encrypted KKZ) or
		decrypted *KKZ = ded*KKoY(encrypted *KKZ)
		Case 2: (*)KK with notarization
		If the NOS field is present, then the (*)KK was notarized before transmission. In this case, the (*)KK subfield contents are processed using the following equations:
	(a)	Compute (*)KN using the contents of the ORG, RCV and CTP fields and the process defined in Section 7.5.
	(b)	Decrypt the contents of the (*)KK field to form (*)KKZ using the equation:
		decrypted KKZ = dKN(notarized KKZ) or
		decrypted KKZ = ded*KN(notarized KKZ) or
		decrypted *KKZ = ded*KN(notarized *KKZ)
7.	(*)KKU	(CKT environment only) (When present)
		Using the rules of Section 8.5, parse the field to obtain the (*)KK; and the P, IDK1 and IDK2 subfields, if present. If the IDK2 subfield is not present, the *KK to be used to decrypt the

Process	Field	Action
		key in the (*)KKU subfield is the only one shared by the message recipient and the CKT.
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If on decryption, the key does not conform to the specification for odd parity, a ESM shall be generated in response to the KSM, with a "K" in the ERF field. I.e., ERF/K Further processing of the message shall cease.
	IDK1	If a (*)KK is received as a (*)KKU in the message, and no IDK1 field is present, then, immediately on decryption of the new (*)KK, that key shall be placed into use for all subsequent processing.
	IDK2	The IDK2 subfield (if present) defines the *KK (*)KKY to be used in decrypting the (*)KKU. Otherwise, the *KK to be used is implicitly defined and is a *KK shared with the CKT. See Section 8.5. If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of a ESM with an "WI" in the ERF field, i.e., ERF/I If a (*)KKU is received, an associated count shall be established for key offsetting the (*)KKU when other keys are received which are encrypted using this (*)KKU. The count (CTP) is initially set to one, and the value one shall be used to key offset decrypt the KD which is received in this message. Decryption of the (*)KKU Let *KKY be the *KK shared between Party B and the center. Let (*)KKZ be the (*)KK that has been received by Party B, the ultimate recipient. The (*)KKU shall be decrypted using the following equations:
	(a)	Compute *KN using the (*)KKY and the contents of the ORG, RCV and CTB fields and the process defined in Section 7.5, above.

Process	Field	Action
	(b)	Decrypt the (*)KKU to obtain the (*)KKZ using the equations: decrypted KKZ = ded*KN(notarized KKU) or decrypted *KKZ = ded*KN(notarized *KKU)
8.	KD	(not used in a CKD environment) At least one and at most two KDs shall be received in a KSM as KD field(s). If a new (*)KK is sent in the KSM, then the KDs shall be decrypted using that key. If the fields do not have subfields, proceed to Case 1. Parse the field(s) to obtain the KDs; and the "P", IDK1 and IDK2 subfields if present.
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If on decryption, the key does not conform to the specification for odd parity, a ESM shall be generated in response to the KSM, with a "K" in the ERF field. I.e., ERF/K Further processing of the Cryptographic Service Message shall cease.
	IDK1	If present, this subfield names the KD. See Section 8.5.
	IDK2	The (*)KK used to decrypt the KD is identified by the IDK2 subfield if present. See Section 8.5. If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field, i.e., ERF/I If an IDK2 subfield is not present, the (*)KK used to decrypt the received (*)KK is the only one shared by the message originator and recipient.
		Decryption of KDs:

Process	Field	Action
		Case 1: The KDs are encrypted by a (*)KK and are not given notarization protection (i.e., there is no NOS field in the KSM or a (*)KK or (*)KKU is received).
		Let KDI be the key received in the KD field. Let (*)KKY be the key encrypting key to be used. Then the KDs are decrypted using the equations:
	(a)	Use the procedure of Section 7.4 and the value of CTX to compute the (*)KKoY:
		where CTX is CTP in a point-to-point environment if no (*)KK is sent in the message and (1) if a new (*)KK is sent. In a CKT environment CTX shall equal (1).
	(b)	Decrypt the KDI:
		decrypted KDI = dKKoY(encrypted KDI) or
		decrypted KDI = ded*KKoY(encrypted KDI)
		Case 2: (only in a point-to-point environment)
		There is no (*)KK field in the Cryptographic Service Message and the KDs were notarized. In this case, the NOS field was included in the KSM and the KD contents of the KD fields shall be decrypted using the following equations:
	(a)	Compute (*)KN using (*)KKY and the contents of the ORG, RCV and CTP fields and the process defined in Section 7.5.
	(b)	Decrypt the KDI using the equation:
		decrypted KDI = dKN(notarized KDI) or
		decrypted KDI = ded*KN(notarized KDI)
9.	KDU	(CKD or CKT environment only)
		If and only if no (*)KK is received as a (*)KKU, then at least one and at most two notarized KDs shall be received as KDU fields.
		Using the rules of Section 8.5, parse the fields to obtain the KDs; and the "P", IDK1 and IDK2 subfields if present.
		Parse the fields to obtain the KDUs; and the "P", IDK1 and IDK2 subfields if present.

Process	Field	Action
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If on decryption, the key does not conform to the specification for odd parity, a ESM shall be generated in response to the KSM, with a "K" in the ERF field. I.e.,
		ERF/K
		Further Cryptographic Service Message processing shall cease.
	IDK1	If present, this subfield names the KD. See Section 8.5.
	IDK2	The *KK used to decrypt the KD is identified by the IDK2 subfield (if present). See section 8.5. If the IDK2 subfield is not present, the *KK to be used for decryption of the KDU is the only *KK shared between the recipient and the center.
		If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field, i.e.,
		ERF/I
	Decryption of KDUs:	
		KDUs shall be decrypted using the processes that follow.
	(a)	Compute *KN using the *KK shared with the center, the contents of the ORG, RCV and CTB fields and the process defined in Section 7.5.
	(b)	Decrypt the KDU using the equation:
		decrypted KD = ded*KN(notarized KD)
10.	IV	(When present)
		If the first character is "E", then the IV that follows is encrypted. If the first character is a "P", the IV does not require decryption before use.
		If an IV is received in encrypted form, the IV shall be decrypted using the equation:
		decrypted IV = dKD(IV)
		where the KD is received in the Cryptographic Service Message.

Process	Field	Action
		If only one KD is received in the in the message, that KD shall be used to decrypt the IV. If two KDS are received, the second shall be used to decrypt the IV. If the first letter is other than "E" or "P"; or the remaining characters are not members of the set, (0-9), (A-F), then an error condition exists. An ESM shall be generated and sent to the originator of the message with an "F" in the ERF field. I.e., ERF/F Further processing of the Cryptographic Service Message may continue prior to ESM transmission.
11.	EDK	(When present) The EDK, if received, is the date and time on which the KDs received in the message shall be placed in use.
12.	CTB	(only used in a CKD or CKT environment) The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of the CTB field. If the field is CTB/b, the CTB value is "b". If a CTB error is detected, a ESM shall be generated to notify the originating party of the CTB error condition. The ESM shall have a "B" in the ERF field, i.e., ERF/B Further processing of the KSM may continue.
13.	CTP	(only used in a point-to-point environment) The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of the CTP field. If the field is CTP/p, the CTP value is "p". If a CTP error is detected, a ESM shall be generated to notify the originating party of the CTP error condition. The ESM shall have a "P" in the ERF field, i.e., ERF/P Further processing of the KSM may continue.

Process	Field	Action
14.	MAC	Compute a MAC from the message.
		The MAC is always computed using the KDs received in the message. If only one KD is received, KDJ, then that key shall be used. If two KDs are received, KDH and KDI, then the key, KDJ (used to authenticate the Cryptographic Service Message), is derived from the equation:
		KDJ = (KDH + KDI)
		The MAC is then:
		aKDJ(MCL/KSMb...bCTX/xb)
		where CTX is CTP and x is p in a point-to-point environment; and CTX is CTB and x is b in a CKD or CKT environment.
		If the computed MAC does not equal the received MAC, the message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include an "M". I.e.,

ERF/M

10.6 Processing A Request For Service Message

Request For Service Messages (RFS) are only received by a CKT. Responses to the RFS are either a RTR if there are no errors detected in the RFS or a ESM if errors are detected in the RFS. A RFS shall be processed as follows:

Process	Field	Action
1.	MCL	Examine the contents of this field to determine the Cryptographic Service Message class.
2.	RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient.
		If RRRR is not the identity of the party receiving the RFS for processing, then the message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
3.	ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
4.	IDU	This field contains the identity of the ultimate recipient. If the field is IDU/UUUU, the ultimate recipient is UUUU.

Process	Field	Action
		The field contents shall be used in selecting the (*)KK to be used in generating the RTR that responds to this Cryptographic Service Message. Further processing of the Cryptographic Service Message may be performed. If the identity of the ultimate recipient is not known to the CKT, a ESM shall be generated and sent to the originator with a "U" in the ERF field. I.e., ERF/U
5.	(*)KK	(When present) Using the rules of Section 8.5, parse the field to obtain the (*)KK; and the P, IDK1 and IDK2 subfields if present. If an IDK2 subfield is not present, the *KK used to decrypt the received (*)KK is the only one shared by the message originator and recipient (center).
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If on decryption, the key does not conform to the specification for odd parity, a ESM shall be generated in response to the RFS, with a "K" in the ERF field. I.e., ERF/K Further processing of the Cryptographic Service Message shall cease. If the key parity is correct, the "P" shall be inserted in the "P" subfield of the (*)KKU field in the RTR generated in response to this Cryptographic Service Message using the rules of Section 8.5.
	IDK1	If present, the IDK1 shall be inserted in the IDK1 subfield of the (*)KKU field in the RTR generated in response to this Cryptographic Service Message, using the rules of Section 8.5. Otherwise, no IDK1 subfield shall be sent in the (*)KKU field in the RTR.
	IDK2	If an IDK2 is present and the IDK2 is not known to the CKT, this shall cause the processing of the RFS to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field, i.e.,

Process	Field	Action
		ERF/I
		Decryption of (*)KKs
		If a new (*)KK is received, then the decrypted (*)KK is computed using the following equation:
		Let (*)KKZ be the key to be decrypted and *KKY be the key encrypting key shared between the CKT and the originator of the RFS.
		Use the procedure of Section 7.4 and CTA to compute the *KKoY.
		Decrypt (*)KKZ
		decrypted KKZ = ded*KKoY(encrypted KKZ) or
		decrypted *KKZ = ded*KKoY(encrypted *KKZ)
6.	KD	At least one and at most two KDs shall be received in a RFS as KD fields.
		If a new (*)KK is sent in the RFS, then only one KD shall be received in the RFS and that KD shall be decrypted using the received (*)KK.
		If the KD fields do not have subfields, proceed to Case 1.
		Parse the field(s) to obtain the KD; and the "P", IDK1 and IDK2 subfields if present. If the IDK2 subfield is not present, the *KK used to decrypt the KD is the only *KK shared by the message originator and the CKT.
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If on decryption, the key does not conform to the specification for odd parity, a ESM shall be generated in response to the RFS, with a "K" in the ERF field. I.e.,
		ERF/K
		Further processing of the Cryptographic Service Message shall cease.

Process	Field	Action
		If the key parity is correct, the "P" shall be inserted in the "P" subfield of the RTR generated in response to this Cryptographic Service Message using the rules of Section 8.5.
	IDK1	If present, this subfield names the KD. The IDK1 shall be inserted in the IDK1 subfield of the KDU field in the RTR generated in response to this Cryptographic Service Message, using the rules of Section 8.5.
	IDK2	The *KK used to decrypt the KD is identified by the IDK2 subfield if present. See Section 8.5.
		If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the RFS to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field, i.e.,
		ERF/I
		Decryption of the KDs
		Case 1 The RFS contains a (*)KK field
		One and only one KD shall be received in the RFS in the KD field. That KD shall be used to authenticate the RFS and to generate the MAC for inclusion in the RTR that responds to this RFS.
		Let KDI be the key received in this field. Let (*)KKY be the key encrypting key sent in the Cryptographic Service Message. The KD is decrypted using the equations:
	(a)	An offset of zero is used to compute the (*)KKoY.
	(b)	Decrypt the KDI using the equation:
		decrypted KDI = dKKoY(KDI) or
		decrypted KDI = ded*KKoY(KDI)
		Case 2: There is no (*)KK field in the Cryptographic Service Message.
		At least one and at most two KDs shall be received in a RFS as KD field(s).
		The KDs are decrypted by a *KK shared with the message originator, and the following equations are used:

Process	Field	Action
	(a)	Use the value of CTA and the procedure of Section 7.4 to compute the *KKoY
	(b)	Decrypt the KDI using the equation: decrypted KDI = ded*KKoY(KDI)
7.	CTA	The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of the CTA field. If the field is CTA/a, the CTA value is "a". If a CTA error is detected, a ESM shall be generated to notify the originating party of the CTA error condition. The ESM shall have a "A" in the ERF field, i.e., ERF/A Further processing of the RSM may continue.
8.	MAC	Compute a MAC from the message. The MAC shall always be computed using the KDs received in the Cryptographic Service Message. If one KD is received, that KD, KDJ, shall be used. If two KDs are received, KDH and KDI, then the key, KDJ (used to authenticate the Cryptographic Service Message), is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ(MCL/RFSb...bCTA/ab) If the computed MAC does not equal the received MAC, the Cryptographic Service Message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include an "M". I.e., ERF/M

10.7 Processing A Request Service Initiation Message

Request Service Initiation Messages (RSI) are received in all environments in order to request that keys be generated and sent to the originating party in a subsequent KSM.

In the CKD environment, a RSI received by the CKD shall result in a RTR being sent to the originating party (see Figures VII and VIII).

In a CKT environment the originating party is expecting that the recipient shall send a RFS to the CKT containing keys to be translated prior to sending the keys (via the originator of the RFS) to the originator of the RSI in a KSM (see Figures IX and X).

If an error in the RSI is detected by the recipient, then a ESM shall be returned.

Request Service Initiation Messages shall be processed as follows:

Process	Field	Action
1.	MCL	Examine the contents of this field to determine the Cryptographic Service Message class.
2.	RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the RSI for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
3.	ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
4.	IDU	(Required only in a CKD environment when the RSI has been received by a center) This field contains the identity of the ultimate recipient. If the field is IDU/UUUU, the ultimate recipient is UUUU. The field contents shall be used in selecting the (*)KK to be used in generating the RTR that responds to this Cryptographic Service Message and in routing the Cryptographic Service Message. If this identity is not known, a ESM shall be sent with a "U" in the ERF field. I.e., ERF/U Further processing of the Cryptographic Service Message may continue prior to ESM transmission.
5.	IDC	(Required in all RSIs from Party B to Party A in a center environment only) This field contains the identity of the CKD or CKT. If the field is IDC/CCCC, the center is CCCC. The field contents shall become the contents of the RCV field in the RFS or RSI generated in response to this Cryptographic Service Message.

Process	Field	Action
		If the identity of the CKD or CKT is not known to the recipient, a ESM shall be generated and returned to the originator with a "D" or "T" in the ERF field for a CKD or CKT environment, respectively. Further processing of the Cryptographic Service Message may continue prior to ESM transmission.
6.	SVR	Parse the field to obtain the designators for the types of service requested. These service types shall be utilized in generating the KSM or RTR that responds to this RSI. Note that a single data key is implicitly requested by the presence of an SVR field. See Table II for a definition of the types of service requested.
7.	EDC	(When present) If this option is not implemented, a ESM may be generated and returned to the originator with an "O" in the ERF field. I.e., ERF/O Alternatively, the field may be disregarded and message processing may proceed. Compute an EDC from the message using the data key for EDC computation (KDX = 0123456789ABCDEF). E.g., EDC = aKDX(MCL/RSIb...bSVR/*KK.KD.IVb) If the computed EDC does not equal the received EDC, there is an error (possibly a transmission error). A ESM shall be generated and returned to the originator. The ERF field shall include an "X". I.e., ERF/X

10.8 Processing A Response Service Message

Response Service Messages (RSM) are received as an authenticated acknowledgement of a DSM, a KSM or a unsolicited RTR (CKD environment only) and shall be processed as follows:

Process	Field	Action
1.	MCL	Examine the contents of this field to determine the Cryptographic Service Message class.
2.	RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient.

Process	Field	Action
		If RRRR is not the identity of the party receiving the RSM for processing, then the Cryptographic Service Message has been misrouted and further processing shall cease. The routing problem shall be resolved outside of this protocol.
3.	ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
4.	IDC	(Used in a CKD or CKT environment when responding to a KSM) Process the value of the IDC field. If the field is IDC/CCCC, the value is CCCC. This is the identity of the CKD or CKT, and is used to select the keys and other data used to process the Cryptographic Service Message. If two RSMs are received from the same originator in response to two KSMs from that originator that used two different centers (CKD or CKT), the identity of the center used is needed to identify the KSM to which the RSM responds. If this identity is not known, a ESM shall be sent with a "D" (CKD environment) or a "T" (CKT environment) in the ERF field. Further processing of the Cryptographic Service Message shall cease.
5.	IDD	(When present) If no IDD field is present, this RSM is in response to a KSM or a unsolicited RTR (CKD environment only). If an IDD field is present, this RSM is in response to a DSM. If the content of the IDD field is null, this indicates that the keying relationship shall be discontinued. Otherwise, each IDD field contains the identity of a discontinued KK, *KK or KD. If the IDD does not match one of the IDD fields sent in the DSM to which this RSM responds, this shall cause processing of the RSM to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field. I.e, ERF/I
6.	IDU	(In response to a unsolicited RTR; CKD environment only)

Process	Field	Action
		If the field is IDU/UUUU, the ultimate recipient is UUUU. If the identity of the ultimate recipient is not known to the CKD, a ESM shall be generated and sent to the originator with a "U" in the ERF field. I.e., ERF/U Further processing of the RSM may continue prior to transmission of the ESM.
7.	MAC	Compute a MAC from the Cryptographic Service Message. The MAC is always computed using the KDs used to generate the MAC in the DSM, KSM or RTR to which the RSM responds. If one KD is received, that KD (KDJ) shall be used. If two KDs were used (KDH and KDI), then the key, KDJ (used to authenticate the Cryptographic Service Message), is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ(MCL/RSMb...bIDD/KKKKb) responding to a DSM aKDJ(MCL/RSMb...bIDC/CCCCb) responding to a KSM in a center environment aKDJ(MCL/RSMb...bORG/OOOOb) responding to a KSM in a point-to-point environment aKDJ(MCL/RSMb...bIDU/UUUUb)responding to a unsolicited RTR in a CKD environment Input to the authentication algorithm starts with the first character following the open parenthesis, "(" of the Cryptographic Service Message and continues through the blank, "b" preceding the MAC field. If the computed MAC does not equal the received MAC, the Cryptographic Service Message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include a "M". I.e., ERF/M Following receipt of a RSM that responds to a DSM, the KD used to authenticate the message shall be discontinued.

10.9 Processing A Response To Request Message

A Response To Request Message (RTR) is received from a CKD or a CKT (see Figures VII-X). A correct RTR shall result in a KSM being generated and sent to the party identified in the IDU field. If an error is detected in the RTR, a ESM shall be generated and returned to the center that originated the RTR.

Response To Request Messages shall be processed as follows:

Process	Field	Action
1.	MCL	Examine the contents of this field to determine the Cryptographic Service Message class.
2.	RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the RTR for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
3.	ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
4.	IDU	Process the value of the IDU field. If IDU/UUUU, the ultimate recipient is UUUU. The field contents shall be used as the contents of the RCV field in the following KSM. If the identity of the ultimate recipient is not known to the recipient, a ESM shall be generated and sent to the center with a "U" in the ERF field. I.e., ERF/U Further processing of the RTR may continue prior to transmission of the ESM.
5.	(*) KKU	(CKT environment only) (When present) The field contents shall be used as the contents of the (*)KKU field in the KSM that is generated in response to this Cryptographic Service Message.
6.	KD	(CKD environment only) At least one and at most two KDs shall be received in a RTR as KD field(s). The *KK used to decrypt the KD is a *KK shared with the CKD.

Process	Field	Action
		Parse the field(s) to obtain the KD; and the "P", IDK1 and IDK2 subfields if present.
	Optional Subfields	
	P	If the "P" subfield is present, the plaintext key shall conform to the specification for odd parity. If on decryption the key does not conform to the specification for parity, further processing of the Cryptographic Service Message shall cease, and a ESM shall be generated in response to the RTR, with a "K" in the ERF field. I.e.,
		ERF/K
	IDK1	If present, this subfield names the KD. See Section 8.5.
	IDK2	The *KK used to decrypt the KD is identified by the IDK2 subfield if present. See Section 8.5. If the IDK2 subfield is not present, the *KK used for decryption is the only *KK shared by the message recipient and the CKD.
		If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the RTR to cease and the generation and transmission to the originating party of a ESM with an "I" in the ERF field, i.e.,
		ERF/I
		Decryption of KDs:
		Let KDI be the key received in the KD field.
	(a)	Compute *KN using the contents of the RCV, IDU and CTA fields and the process defined in Section 7.5.
	(b)	Decrypt the KDI:
		decrypted KDI = ded*KN(notarized KDI)
7.	KDU	The field contents shall be used as the contents of the KDU field in the KSM that is generated in response to this Cryptographic Service Message.
8.	IV	(CKD environment only) (When present)
		The IV is generated by the CKD. On receipt of the RTR, the IV is stored for retransmission to the ultimate recipient and decrypted for use by the recipient using the process given below.

Process	Field	Action
		Process the contents of the field. If the first character is "E", then the IV that follows is encrypted. If the first character is a "P", the IV does not require decryption before use. If the first letter is other than an "NE" or "P"; or the remaining characters are not members of the set, (0-9), (A-F), then an error condition exists. A ESM shall be generated and sent to the originator of the Cryptographic Service Message with an "F" in the ERF field. I.e., ERF/F Further processing of the Cryptographic Service Message may continue prior to transmission of the ESM. Case 1: Encrypted IV If an IV is received in encrypted form, the IV shall be decrypted as follows: Let KDH be the KD received in the Cryptographic Service Message. If two KDs are received, the second KD shall be used as the KDH. decrypted IV = dKDH(encrypted IV) Case 2: Plaintext IV If an IV is received in plaintext form, then it does not require further processing.
9.	EDK	(CKD environment only) (When present) This field contains the date and time that the KDs become effective. The EDK shall be used by the recipient and be resent to the ultimate recipient. The field contents shall be used by the recipient and as the contents of the EDK field in the KSM that is generated in response to this Cryptographic Service Message.
10.	CTB	Process the value of the CTB field. If the field is CTB/b, the CTB value is "b". This value (i.e., "b") shall be sent as the contents of the CTB field in the KSM that is generated in response to this RTR.
11.	CTA	(CKD environment only)

Process	Field	Action
		The general rules for handling of counters are given in Sections 7.3.2 and 7.3.3. Process the value of CTA. If the field is CTA/a, the CTA value is "a". If a CTA error is detected, a ESM shall be generated to notify the originating party of the CTA error condition. The ESM shall have a "A" in the ERF field, i.e., ERF/A Further processing of the RTR may continue.
12.	MAC	Compute a MAC from the Cryptographic Service Message. In a CKT environment, when a (*)KK is returned in the Cryptographic Service Message, the KD used to authenticate the Cryptographic Service Message shall be the KD sent in the ERS or RFS to which this message responds. Otherwise, the MAC is always computed using the KDs received in the message. If one KD is received, KDJ, then that key shall be used. If two KDs are received (KDH and KDI), then the key, KDJ (used to authenticate the Cryptographic Service Message), is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ(MCL/RTRb...bCTA/ab) for a CKD environment or aKDJ(MCL/RTRb...bCTB/bb) for a CKT environment If the computed MAC does not equal the received MAC, the Cryptographic Service Message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include an "M". I.e., ERF/M

11. References

1. American National Standard X3.4-1977, Code for Information Interchange.

2. American National Standard X3.92-1981, Data Encryption Algorithm.

3. American National Standard X3.106-1983, Modes of Operation of the DEA.

4. American National Standard X9.9-1982, Financial Institution Message Authentication (Wholesale).

5. NBS Special Publication 500-20-Revised 1980, Validating the Correctness of Hardware Implementations of the NBS Data Encryption Standard.

6. NBS Special Publication 500-61-1980, Maintenance Testing for the Data Encryption Standard.

APPENDICES

These Appendices are included for information only and are not part of the standard.

APPENDIX A

EXAMPLES OF KEY DISTRIBUTION AND CONTROL

Management of Manually Distributed Key

A.1 General

This section recommends minimum management and control requirements for any key that is manually generated, shipped, stored, used or destroyed.

A.2 Appointment of Encryption Personnel

A.2.1 Personnel responsible for key management should be designated by the management of the sending and receiving financial institutions.

For example, management should designate Key Management custodians (hereinafter referred to as custodians) and alternate custodians as needed.

A.2.2 No one person should ever have overall control of encryption material and physical keys for encryption devices.

A.2.3 At least two custodians should be available for each operational shift, where required. For example, personnel assigned these responsibilities should be present for carrying out their duties within sixty minutes after notification of a non-planned need to change keys.

A.2.4 Any alternate custodian can perform the duties of a custodian in the absence of a regularly assigned custodian. Note that to provide proper coverage for multiple-shift operations, management may find it useful to designate personnel at subsidiary levels, such as, "Key Inserters", "Key Controllers", etc.

A.3 Responsibilities of custodians

A.3.1 Responsibility for the receipt, verification of contents, and storage of keying material, should be under the dual control of two custodians.

A.3.2 Custodians should:

(1) receive and store keying material;

(2) maintain a record of personnel authorized to utilize keying material;

(3) verify, in clear view of an alternate custodian, that the received new keying material is intact;

(4) witness the destruction of the old keying material;

(5) prepare a record of keying materials used;

(6) enter and change keying materials; and

(7) destroy previously used keying materials when directed and in the presence of an alternate custodian who should witness the destruction.

A.3.3 Two custodians should control the physical key(s) which allow(s) entry to the encryption or authentication device.

A.4 Shipment of Keying Material

A.4.1 Keying material to be shipped should be packaged under dual control of the originator's custodians. These materials should be shipped under dual control with return receipt requested. Registered mail or a courier (preferably an employee) may be used.

For example, the originator's custodians package the material along with a transmittal form. The transmittal form could describe and give the quantity of materials shipped. These materials could be shipped by registered mail, after wire advice to all recipient's custodians of the date of shipment.

A.5 Receipt of Keying Material

A.5.1 Under dual control, the custodians should upon receipt of keying material, verify the contents by counting the number of packages before checking the manifest and determine if the material has been damaged in any way.

For example, keying material could be delivered wrapped in plastic or an opaque envelope. This should be inspected for signs of damage or tampering. A comparison of the identifying data should be made with the information contained on the transmittal form.

A.5.2 The transmittal form should be signed by each receiving custodian. One copy should be returned to originator and one retained on file.

A.5.3 The originator and local security officer should be notified immediately if there is a question regarding the integrity of the keying material. In the event of notification, the originator in consultation with the recipient should make a determination as to whether to use the material.

Transfer of keying material should be accomplished under dual control and at a mutually acceptable time. Transfer of control should only be to properly designated individuals and should be documented.

A.6 Storage of Keying Material and Encryption/Authentication Device Physical Keys

A.6.1 All keying material under control of the custodians should be stored in a secure receptacle, such as a lock box or safe.

A.6.2 The receptacle should have two locks, and access to it should require the key(s) (or combination code of locks) of both custodians.

A.6.3 A usage record of the contents of the receptacle should be maintained by the Custodians.

For example, a log should be maintained recording the deposits and withdrawals for each access to the secure receptacle.

A.6.4 The receptacle should be under control of the appropriate custodians per operational shift (if required).

A.6.5 The receptacle should be kept in a secure location. For example, a security or guard control room or any other equally secure area may be designated as the repository for the receptacle.

A.6.6 Backup physical keys (or combination code of locks) for the locked receptacle should be kept in a secure location. Access to such keys should require the presence of the custodians.

A.6.7 Physical keys for the encryption or authentication devices should be controlled by the custodians. Where the equipment has two locks, each custodian should have access to only one key.

A.6.8 Physical keys for the encryption or authentication device should be kept in a secure location.

For example, the primary set of keys should be kept in the dual locked receptacle while the backup keys should be kept in a similar dual locked receptacle located in a secure area such as a security or guard control room.

A.6.9 Due to the criticality of keys, short term power outages should not result in a loss of key.

A.7 Use of Keying Material

A.7.1 Whenever it is necessary to change key, the materials should be removed from the locked receptacle(s) by the custodian(s). The next key or key component (e.g., page) should be removed by the custodian and an accompanying Disposition Record (e.g., contained in the front section of the key variable book) should be completed.

A.7.2 The custodian should not be aware of or familiar with the contents of the key component which is not that custodian's responsibility to insert.

A.7.3 The current key component, which is presently in use, should be sealed in an envelope by the custodian and placed back into the locked receptacle after loading and testing the encryption device to ensure proper loading of the key component.

A.7.4 A supplemental Disposition Record should be kept to record the return of the key component information to storage and subsequent reissue if required.

A.8 Destruction of Keying Material

A.8.1 Keying material should not be destroyed by the custodian until instructed to do so.

A.8.2 Keying material (e.g., pages) should be destroyed by either crosscut shredding or any other method which destroys it beyond reconstruction.

A.8.3 The custodian (or alternate custodian, as the case may be) should witness the destruction of the keying material without becoming aware of its contents.

A.8.4 A record of destruction should be maintained. For example, this record should contain sufficient information to identify the material destroyed (e.g., the keying material book number, keying material page number), signature of person destroying the page (custodian), date/time, and signature of a witness.

A.8.5 A Destruction Record should be stored in the locked receptacle containing the keying material in a previously designated secure location.

A.9 Archival Storage of Keys

Where it is necessary that keying material be stored for archival purposes,

That keying material should be handled in accordance with the requirements of Sections A.5 and A.6 if the confidentiality of the data or secrecy of the key or both are required. An alternative is to protect the keys by encryption in a key used only for archival storage. In this case, only the storage key should be protected in accordance with this section.

Where keying material must be stored for archival purposes and that material has no further security value (security life), it may be stored using the usual procedures in place for the storage and protection of vital records.

APPENDIX B

EXAMPLE OF MANUAL KEY DISTRIBUTION

This Appendix presents a method to enhance security when distributing a key encrypting key for communications between two parties. In this example, dual control is maintained by splitting knowledge of the manually distributed key between two parties rather than between two individuals working for the same party.

Assume that a custodian representing each of the two parties (e.g., Bank A and Bank B) is assigned the task of establishing the key encrypting key to be used between the two banks.

Each custodian generates a DEA key component consisting of 56 random bits and eight odd parity bits.

A courier from Bank A then takes Bank A's generated key component to Bank B. Likewise, a courier from Bank B takes Bank B's generated key component to Bank A. At each bank, the bank's custodian and the courier from the other bank enter their key components into the key management facility, each person taking care not to disclose his or her key component to the other person. The device accepts the entries, checks parity on each key component and then performs an exclusive-or operation on the two key components and converts the result to odd parity. The resulting key is the manually entered key encrypting key shared between the two banks.

An example of the derivation of this key encrypting key follows:

Bank A's Key component = 1111 0100(F4) 1101 0101(D5) 0010 1001(29)
1000 1111(8F) 0000 1110(0E) 0011 0111(37)
1100 0010(C2) 1001 0001(91)

Bank B's Key component = 1101 0000(D0) 0001 0101(15) 1011 0101(B5)
1011 0110(B6) 1011 1001(B9) 1001 0111(97)
1010 0100(A4) 0000 1101 (0D)

Resulting Key = 0010 0101(25) 1100 0001(C1) 1001 1101(9D)
0011 1000(38) 1011 0110(B6) 1010 0001(A1)
0110 0111(67) 1001 1101(9D)

APPENDIX C

PSEUDORANDOM KEY & IV GENERATOR

C.1 Purpose

The purpose of this Appendix is to present an example of a Pseudorandom key and IV generator.

C.2 Algorithm

Let ede*X(Y) represent the DEA multiple encryption of Y under key *X. Let *K be a DEA key pair reserved only for the generation of other keys, let V be a 64-bit seed value which is also kept secret, and let + be the exclusive-or operator. Let DT be a date/time vector which is updated on each key generation. I is an intermediate value. A 64-bit vector R is generated as follows:

$$I = ede^*K(DT)$$

$$R = ede^*K(I + V)$$

and a new V is generated by $V = ede^*K(R + I)$.

To obtain a DEA key, every eighth bit is reset to odd parity. To obtain an Initialization Vector (IV), R may be used directly.

APPENDIX D

DESIGN OF CRYPTOGRAPHIC EQUIPMENT

D.1 Physical Protection of Cryptographic Equipment

Physical protection includes the mounting mechanism accessible only from the interior of the locked equipment, institution of management controls, use of software security features, etc.

D.1.1 Equipment Enclosure

Cryptographic equipment enclosures should be designed such that two physical locks should be operated in order to disassemble the equipment to an extent that would permit undetectable access to internal circuitry. The two locks needed for operations of the device need not be the locks needed for access to the circuitry.

All holes placed in the outside surface of the equipment during manufacture should be located such that undetectable access to all storage and processing circuitry, as well as undetectable disassembly of the equipment, are not possible using these holes.

D.1.2 Protection Against Unauthorized Use

For periods during which the cryptographic equipment is required to be dormant, a means (e.g., a battery) should be provided to make the cryptographic equipment inoperable without loss of currently active keys.

D.2 Transfer Operation

Electronic Key Transfer into cryptographic equipment from a key loader is initiated by the cryptographic equipment under the dual control of operating personnel.

D.3 Key Management Functions & Alarms

D.3.1 DEA Devices

Method 1

Two DEA devices should be used to do the same encryption of plaintext data. Their output should be compared. Any difference between the output should generate an alarm and should cause the cipher text output immediately to cease until operating personnel eliminate the error conditions, or take such other action as may be prescribed by approved procedures. A means to test the comparator circuits or logic, and the associated inhibiting circuits or logic automatically should be provided.

Method 2

(1) S-Box Test

This test consists of loading one or more known keys (test variables) and one or more known 64-bits input into the transmit DEA device and operating the DEA key generator until all S-box entry combinations for each S-box have been applied. The final output is then compared with all 64-bits of the known current result (determined previously, off-line and stored in the equipment).

If they fail to compare, an alarm should be generated automatically and all ciphertext output should be inhibited until operating personnel eliminate the error condition, or take such other action as prescribed by approved operating procedures.

A means of automatically testing the comparator circuits or logic and associated inhibiting circuits or logic (i.e., cause an intentional error) should be provided. (Descriptions of several S-box tests are contained in National Bureau of Standards Special Publications 500-20 and 500-61).

(2) DEA Checkword Test

After the S-box test has been performed, and after a new key is loaded into the cryptographic equipment, a known 64-bit input word is encrypted in the new key using the ECB mode; the resulting 64-bit checkword is stored. This checkword should be retained in storage and used until the new key is superseded.

The DEA checkword test consists of encrypting the known 64-bit input word in the current key and comparing the result with all 64-bits of the checkword. If they fail the comparison test, an alarm should be automatically generated and any authenticated data, ciphertext, or any combination thereof, output of the DEA equipment should be inhibited until operating personnel eliminate the error condition, or take such other action as prescribed by approved operating procedures.

D.3.2 Counters

Cryptographic equipment should have an alarm to detect and report the erasure, loss or lowering of a counter (see Section 7.3).

D.4 Working Key Tests

Where (*)KKs or KDs are computed as the exclusive-or of two or more key components to implement dual control with split knowledge, the parity of resultant (*)KK or KDs should be corrected if necessary. Operation of the device may be inhibited and an alarm condition generated if any key is one of the four self-dual (weak) keys, i.e.,

01	01	01	01	01	01	01	01
FE	FE	FE	FE	FE	FE	FE	FE
1F	1F	1F	1F	0E	0E	0E	0E
E0	E0	E0	E0	F1	F1	F1	F1

D.5 Control Functions

NOTE: It is not necessary to provide individual locks for each control function. They may, for instance, be collected behind a locked cover or gated by a physical key switch.

Cryptographic equipment should provide for the following controls under the conditions listed:

NAME	FUNCTION	CONDITIONS
Power On/Off	Turns primary power (and internal battery) ON or OFF and causes zeroization of critical storage when in the OFF position (see Section D.6).	Optional feature. The use of a lock is recommended.
Standby Mode	Provides the capability to render the DEA device inoperable during unattended periods, without zeroizing the key variable (see Section D.1.2).	Recommended. Should be under control of two locks.
Alarm Reset	Provides the capability to clear alarms after a fault has been corrected by repeating those security checks which could have generated the alarm condition. Performance of the security checks should be successful (i.e., the condition causing the alarm should have been corrected) before the alarm state can be exited. The authenticated or ciphertext output should be inhibited until the alarm state is exited.	Recommended. Should be under control of a lock.
Test Mode	Causes Cryptographic equipment to perform tests.	Recommended. May be under control of a lock.
Lamp Test	Provides assurance that indicators are operable.	Recommended for incandescent lamps. Not required for LEDs. No lock recommended.
Key Component Entry	Provides for external entry of DEA key components, either manually or automatically. (This does not include "downline loading".) Authenticated or ciphertext output should be inhibited during entry of the key components if the DEA key components are automatically placed in a DEA device as a result of entry.	Recommended if external key component entry devices are used. Should be under the control of the locks.

NAME	FUNCTION	CONDITIONS
Bypass	Provides the capability for bypassing the DEA device and transmitting unauthenticated messages when DEA facilities are in an alarm condition or other malfunction condition.	Optional, should be under control of a lock if present.
Secure Mode	Provides capability to transmit and receive ciphertext (key management information).	Optional feature. Should be under the control of locks.
Zeroize	Provides capability to zeroize all unencrypted key components.	Recommended. Although no lock is required, protection should be provided to protect the switch from inadvertent operation.

D.5.1 Status Indicators

Cryptographic equipment should provide for display of the following indication of status under the conditions listed below.

NAME	FUNCTION	CONDITIONS
Power On	Indication that proper electrical power is available for equipment operation.	Recommended.
Bypass	Indication that the equipment is not in authentication or encipher/decipher state.	Recommended.
Test	Indication that cryptographic equipment is in a test mode, as opposed to an operational mode.	Recommended.
Battery	Indication that the internal battery is operating properly and is capable of retaining critical storage.	Recommended.
Alarm	Indication that an error in operation of the cryptographic equipment has occurred or that attempted tampering has been detected. Authenticated or encrypted output should be automatically and immediately disabled when an alarm occurs, if not in the bypass condition.	Recommended.

NAME	FUNCTION	CONDITIONS
Audible Alarm	Same as Alarm	Optional feature. Not a front panel indicator. A dry contact relay type of interface may be used.
Parity	Indication that an error in parity has occurred during key component entry or during internal transfer of the key component. Further internal key component transfers should be inhibited until the condition which caused the error is corrected and the correct key component has been entered.	Recommended.

D.6 Retention of Critical Storage

Critical storage (e.g., key component final storage location(s), counters and test data) in cryptographic equipment should be retained during primary power interruptions. Cryptographic equipment should have a means of determining whether critical storage has been properly maintained during interruption of primary power. Cryptographic equipment should also have the ability to maintain keys for a minimum of 96 hours whenever primary power is interrupted.

D.7 Electromagnetic Interference Design and Construction Practices

Good electromagnetic interference (EMI) design and construction practices should be followed in all aspects of cryptographic equipment design. To provide a level of protection to keying material from compromising emanations through conduction and radiation and to assure that keying material and circuits processing that information are not susceptible to disruption by external transients (e.g., radiation from nearby radio and television stations, airport radars, microwave data transmission links, etc.), the cryptographic equipment should comply with one or more of the following standards:

(a) Title 47, Code of Federal Regulations, Part 15, Subpart J.

(b) Verein Deutcher Electro Techniker 0877 and 0804

(c) U.S. Mil-Std-461b class A-3 equipment and Mil-Std-462 as required by Federal Standard 1027.

Information on construction practices may be obtained from, Interference Technology Engineers' Master, "Basic Shielding Aids," I.T.E.M., 1980.

Note that construction practices recommended by the Interference Technology Engineers' Master include:

(a) use of an enclosure constructed of 16-Gauge ferrous material

(b) an A.C. line filter

(c) ventilation filter

(d) use of copper foil

(e) use of R.F. gasketing material

(f) shielding openings in the enclosure such as windows

APPENDIX E

ERASING (ZEROIZING) RECORDING MEDIA USED

FOR STORAGE OF KEYING MATERIAL

E.1 Purpose

To establish guidance and uniform erasing (zeroizing) (e.g. degaussing, erasing and overwriting), clearing and destruction procedures for storage material used so that unauthorized access to or compromise of the data is prevented.

E.2 General

Due to the physical properties and retentive capabilities of storage media and devices (e.g. magnetic cores, drums, disks and various microelectronic circuits) used to store, record or manipulate sensitive data, special precautions should be taken to safeguard against the compromise of possible residual information. This Appendix presents recommended procedures for such zeroization or destruction.

E.3 Cathode Ray Tube

A display CRT can be considered zeroized if, after visual inspection, it is determined that no sensitive information has been etched into the CRT phosphor coating.

If there is any doubt after inspection of the screen, the CRT surface should be highlighted by filling the screen with vectors to create a raster effect to light up the entire screen. The brightness of the raster can be varied with the intensity control. Any burns or uneven illumination of the phosphor coating that could be considered compromising should then be easily detected. Random burns on the CRT should not necessitate automatic classification of the CRT as containing sensitive information.

Should any area of the CRT be determined to contain sensitive information, the CRT should remain classified at the highest level of residual information.

If the CRT becomes defective and cannot be purged of sensitive information, it should be destroyed.

E.4 Magnetic Core Memory

To zeroize a magnetic core memory, overwrite all data bit locations. All data bit locations should be set to zeros and verified for successful entry; then all locations should be set to ones and the verification repeated. This overwrite procedure (with random hard copy readout or other equivalent verification at conclusion) should be executed alternately with zeros and ones for 1000 cycles. Finally, non-sensitive, arbitrary data should be written in all data bit locations and left in the core.

Alternate Procedures:

(a) The magnetic core memory should be destroyed by pulverizing, melting, incinerating.

(b) Expose all cores to a recommended magnet. The magnet should be held within 1 cm. of each core.

E.5 Disk Pack

To zeroize magnetic disk packs, overwrite all data bit locations three times by setting zeros and ones alternately. Verify successful entry of the overwrites through a random hard copy readout or equivalent verification. Write non-sensitive arbitrary data on all data locations on all tracks of the disk and leave it there.

If the disk has failed in such a way that it cannot be overwritten or the overwrite cannot be verified, clear the disk by exposing the recording surface to a permanent magnet assembly. Cover the magnet assembly with a lintless wiping tissue to prevent damage to recording surfaces.

Wipe the entire surface at least three times with the magnet.

Alternate Procedures:

(a) Apply an emery wheel or sander to the recording surface of an inoperative disk. Ensure that the entire surface is completely removed prior to disposal.

(b) The resin binder and ferric oxide surface can be completely removed/stripped (chemically destroyed) from the disk with concentrated hydrochloric acid (55-58%).

(c) Melting

E.6 Drum

To zeroize a magnetic drum, overwrite all data bit locations three times by setting zeros and ones alternately. Verify successful entry of the overwrites through a random hard copy readout or other equivalent verification.

Write non-sensitive arbitrary data on all data locations, verify that the data have been written to these locations, and leave it there.

If the drum has failed in such a way that it cannot be overwritten or the overwrite cannot be verified, zeroize the drum by exposing the recording surface to a permanent magnet assembly. Cover the magnet with a lintless wiping tissue to prevent damage to the recording surface. Wipe the entire surface at least three times with the magnet. Ensure that all recording areas of the drum are exposed to the active area of the magnet assembly.

E.7 Magnetic Tapes

Magnetic tapes should be zeroized with a degausser. Magnetic tapes may be cleared by overwriting one time with any one character. However, cleared magnetic tapes should be safeguarded, controlled, and marked at the level commensurate with the most sensitive information recorded on them before they were released for destruction. Before release of a zeroized magnetic tape, it should be subjected to two degaussing cycles and removed from the reel, then destroyed by disintegration into pieces 9 mm or smaller, or incineration.

E.8 Internal Memory, Buffers and Registers

Internal memory, buffers, or registers should be zeroized initially by use of a hardware clear switch or power-on/off reset cycle; secondarily by overwriting all data bit locations with continuously changing random data for 1000 cycles. Periodic verification should be made that the method(s) are working correctly. Verify successful entry of the overwrites through a random hard copy readout or through other equivalent verification. Finally, all locations should be overwritten with non-sensitive, random data and verified.

E.9 Semiconductor Memory

(1) Random Access Memory (RAM) should be initialized by use of a power-on/off reset cycle. Overwrite the storage area by alternately setting each data bit location to all zeros then all ones for 1000 cycles. Periodic verification should be made that the method is working correctly. Verification may take the form of random sampling or use of a read and compare program. Finally, all locations should be overwritten with non-sensitive data and verified.

(2) Erasable Programmable Read Only Memory (EPROM) should be initialized by optical ultraviolet erasing the entire array. Zeroization should be verified. All storage locations should be overwritten with non-sensitive random data and verified.

(3) Electrically Alterable Read Only Memory (EAROM) should be initialized by pulsing all gates. Zeroization should be verified. All storage locations should be overwritten with non-sensitive random data and verified.

(4) Electrically Erasable Programmable Read Only Memory (EEPROM) should be initialized by pulsing the erase control gate. Zeroization should be verified. All storage locations should be overwritten with non-sensitive random data and verified.

(5) Read only Memory (ROM) is physically programmed during manufacture. Physical destruction is the only recommended method to ensure erasure.

E.10 Paper Materials

Paper materials should be destroyed by burning, pulverizing, or crosscut shredding. When material is pulverized, all residue should be reduced to pieces 5mm or smaller. When material is burned, the residue should be reduced to white ash.

E.11 Platens and Ribbons

The printer platen and ribbon should be removed from a printer before the printer is released. Platens (only the rubber surface should be physically removed for destruction) and ribbons should be destroyed (e.g., by incineration).

APPENDIX F

WINDOWS AND WINDOW MANAGEMENT

F.1 Purpose

In a Key Distribution Center or Key Translation Center environment it is possible for a recipient to receive messages whose counts are out of sequence, yet the messages are authentic in that the MAC is correct.

A party assuming the role of recipient may establish a window, representing a range of reception counter values, such that the corresponding messages, should they arrive out of sequence, will be accepted without declaring an error. In addition, it is necessary to maintain a record of messages as they are received, i.e., the counter values obtained from received messages are recorded in "window storage".

F.2 Window Size

The lower-bound of the window is defined as the base, b; the upper bound of the window is defined as the top, t. The window size is the top less the base plus one, or, t-b+1. Management of the window is the responsibility of the recipient.

F.3 Detection of Duplicate Messages

When a message is received its count value is first checked to ensure it has not previously been processed. Duplicate messages are rejected with an appropriate error message as described in the body of the standard (see Section 7.3). Otherwise, the messages count value is recorded in window storage.

F.4 Processing Messages Using a Window

If the (origination) count is within the window, as delimited by the base and the top, and has not previously been received, the message is accepted. When the (origination) count is not equal to the base, no adjustment to the window occurs. When the (origination) count corresponds to the base, the window is adjusted such that the base is set to the first unused counter value and the top is established as the sum of the adjusted base and the window size minus one.

If a message is received whose (origination) count falls below the base of the window or if the message count has been previously received (the base being the lowest acceptable count), the message is not accepted and an error is reported to the originator as described in the body of the standard (see Section 7.3).

If a message is received whose (origination) count is above the top of the window, the message is accepted. The window is adjusted so that the count can be recorded as received. The base is adjusted to be the lowest unreceived count such that the received (origination) count is less

than or equal to the top of the window. In some cases, the received count may be below the new base.

Subsequent action on the part of the party receiving the error message is also detailed in the body of this standard (see Section 7.3.3).

A summary of the above action appears in Table F.I.

TABLE F.I
PROCESSING COUNTERS
(MAC's Checked)

Received Count (CTA, CTB) Within Window	Received Count (CTA, CTB) Outside of Window	Received Count (CTA, CTB)		Action to take on receipt of ESM or ERS	
Received count not checked off	Received count checked off (duplicate)	Received count greater than top	Received count less than base	Received count (CTA, CTB) greater than origination (stored) count	Received count (CTA, CTB) less than or equal to origination (stored) count
• Accept msg. • Send RSM • Checkoff count • Log (optional) • If received count equal base, adjust window "up" to first unused count • If received count greater than base do nothing	• Reject msg. • Send ESM with base and received counts • Log (mandatory) • No window adjustment	• Accept msg. • Adjust window "up" • Checkoff count • Send RSM • Log (mandatory)	• Reject msg. • Send ESM with base and received count • Log (mandatory) • No window adjustment	• Set origination count equal to expected count (CTA or CTB) OR • Change (*)KK • Log (mandatory) • Send new KSM or RTR with corrected count OR • Wait for new key request (RFS or RSI)	• Log (mandatory) • Send new KSM or RTR with current origination (stored) count OR • Wait for new key request (RFS or RSI)

APPENDIX G

DUAL CKT APPLICATION

G.1 General

When a party subscribing to one Key Translation Center (CKT1) wishes to establish a keying relationship with another party which does not subscribe to CKT1 but does subscribe to a second Key Translation Center (CKT2), the concept of using a Key Translation Center to establish a key encrypting key relationship between parties can be extended. The dual CKT application allows the initiating party to obtain a keying relationship with any subscriber to the second center.

Note that this augments the existing automated key management architecture by adding a fourth layer; an automatically distributed key encrypting key is used to establish the key encrypting key relationship with the second center. This can be accomplished by an iterative use of the protocol in the CKT environment.

G.2 Process

Two Key Translation Centers are used to establish key encrypting keys (or key pairs) and data keys between Parties A and B which do not share a key encrypting key (or key pair) with each other and do not share a key encrypting key pair with the same Key Translation Center. There are three manually distributed key encrypting key pairs, one shared by the two Key Translation Centers, CKT1 and CKT2; one shared by Party A and CKT1 and the third shared by Party B and CKT2.

In order for Party A to establish a keying relationship with Party B, Party A sends a request for translation of a key encrypting key pair (*KKA2) to his Key Translation Center (CKT1) with CKT2 identified as the ultimate recipient. CKT1 then returns a message to Party A containing *KKA2 encrypted under the manually distributed key encrypting key pair shared by the two centers (CKT1 and CKT2).

Party A sends this encrypted key to CKT2 and thus establishes a key encrypting key pair relationship with CKT2 (the key is *KKA2).

Party A then sends a request for translation of a key encrypting key, (*)KKAB, and a data key, KD, or a request for translation of one or two data keys only, to CKT2 with Party B as the ultimate recipient. If (*)KKAB is sent, (*)KKAB is encrypted under *KKA2, and the KD is encrypted under (*)KKAB. Otherwise the KDs are encrypted under *KKA2.

G.3 Message Flow in a Dual Key Translation Center Environment

Figures G.I-G.II show the flow of Cryptographic Service Messages necessary to establish communications between two parties that have key encrypting key relationships with different Key Translation Centers. The two CKTs must have a key encrypting key pair relationship with each other.

Initially, one of the parties should establish a key encrypting key pair relationship with the other party's CKT. Once this is accomplished, keys may be defined for communication between the two parties according to the steps outlined in Section 8.6.4.

Cryptographic Service Message flow for establishing the key encrypting key relationship with the other party's center should be as follows:

(1) Let Party B be a party that wishes to communicate with Party A, does not currently share a key with Party A, may not have a key generation capability, and does not share a key encrypting key pair with the same CKT as Party A.

A RSI may be sent from Party B to Party A requesting the type of key(s) and (optionally) an IV to be provided. The RSI should also identify the Party B's CKT.

If the RSI received by Party A from Party B contains error(s), then Party A should return a ESM to Party B, and Party B may try sending another RSI to Party A. If Party A has no key generation capability and cannot otherwise acquire keys, then Party A should return a ESM to Party B with an "O" in the ERF field, and secure communications cannot be established.

(2) Let Party A be a party that desires to send keys to Party B with whom there is no commonly shared (*)KK (perhaps in response to a RSI message from Party B).

Party A sends a RFS message to CKT1 containing the *KK to be sent to CKT2 (the ultimate recipient). The *KK is encrypted under a *KK shared between Party A and CKT1. A KD is also sent, encrypted under the *KK sent in the message, and is used to authenticate the Cryptographic Service Message. If an error(s) is detected in the RFS by CKT1, then CKT1 returns a ESM to Party A.

(3) The RFS is received by CKT1. The *KK received in the message should be decrypted using the *KK shared by the center and Party A, and notarized using the identities of Party A and CKT2 and the *KK shared between CKT1 and CKT2 and the count associated with that *KK. The result is inserted in the *KKU field of a RTR returned to Party A. The KD contained in the received RFS is used to authenticate both the RFS and the subsequent RTR. If an error(s) is detected in the RTR received by Party A, a ESM is returned to the Key Translation Center.

(4) Party A sends a KSM to CKT2 containing the *KKU field received in the RTR. Party A also sends a KD encrypted under the *KK sent in the KSM in order to allow the center to authenticate the message. If CKT2 receives the KSM from Party A without error, then a RSM is returned to Party A, otherwise, a ESM is returned to Party A.

Party A may resend a KSM to CKT2 an arbitrary number of times, but Party A does not send a new KSM (i.e., utilizing new keys or a new count for the (*)KK specified in a KSM) until the old KSM is acknowledged by a RSM or a ESM.

(5) At this point, Party A has established a *KK relationship with CKT2. Further establishment of keys between Party A and Party B is accomplished using the procedure of Section 8.6.4.

FIGURE G.I

DUAL KEY TRANSLATION CENTER APPLICATION

(Normal Message Flow

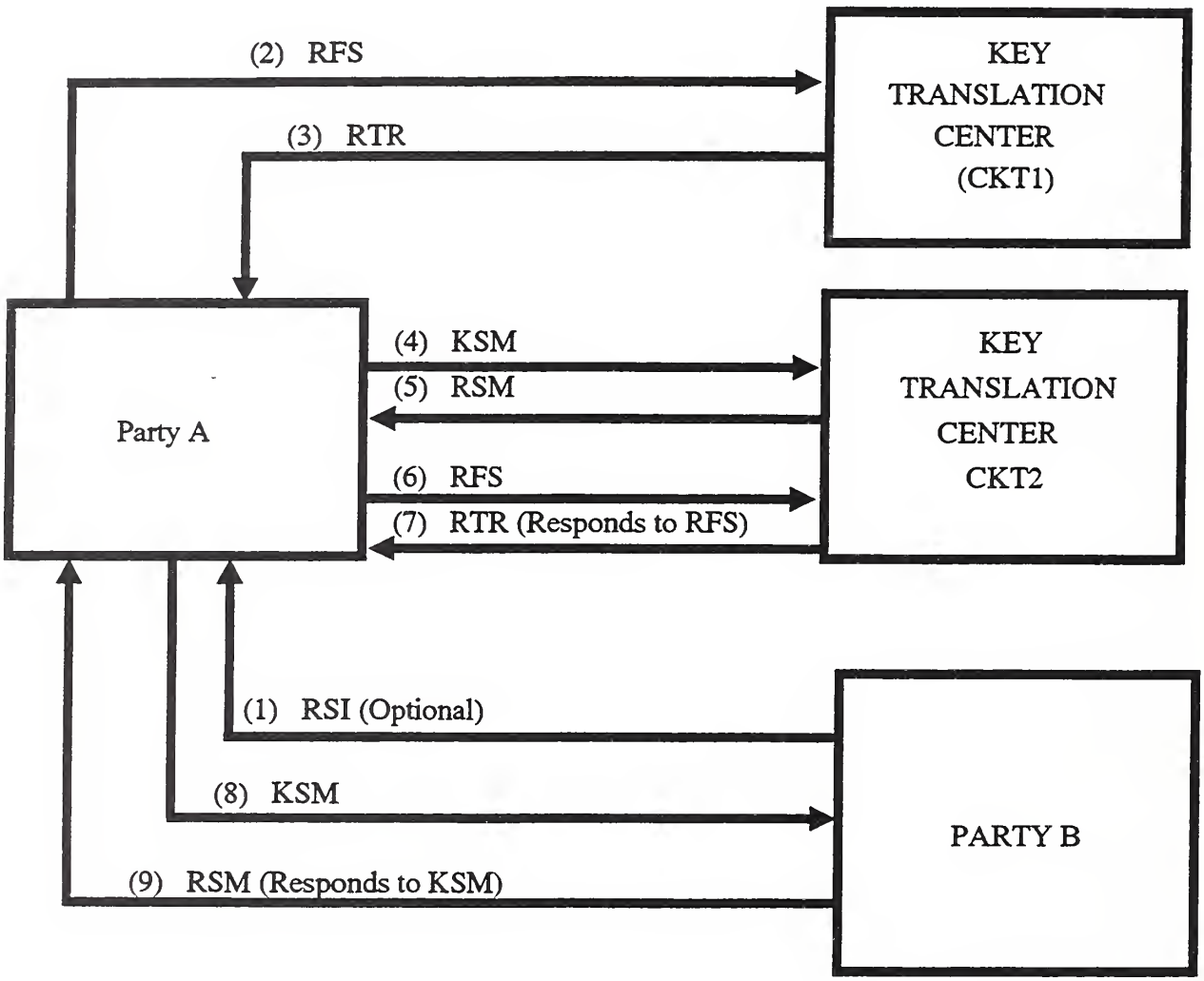
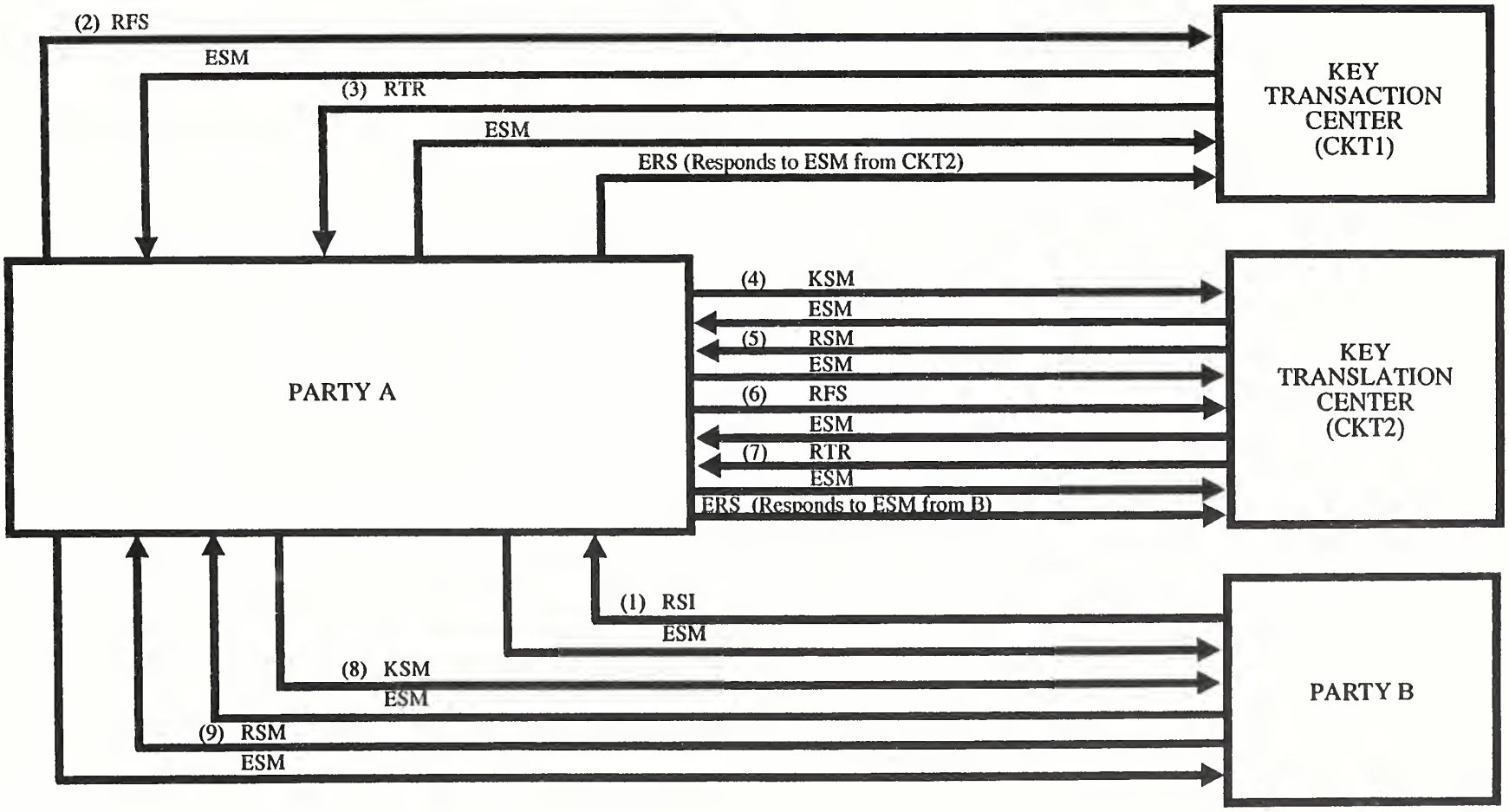


FIGURE G.II

DUAL KEY TRANSLATION CENTER APPLICATION

(Message Flow With Errors)



AMERICAN BANKERS ASSOCIATION
1120 Connecticut Avenue, N.W.
Washington, D.C. 20036